Departamento de Informática e Ingeniería de Sistemas
Universidad Zaragoza

Escuela de Ingeniería y Arquitectura
Universidad Zaragoza

Universidad Zaragoza

Proyecto Fin de Carrera

Generador automático de fichas de personajes para un entorno periodístico

Pilar Blázquez Larraz

Director: Ángel Luis Garrido Marín
Co-director: Óscar Gómez Rubio
Ibercentro Media Consulting & Services

Ponente: Sergio Ilarri Artigas
Departamento de Informática e Ingeniería de Sistemas

Ingeniería Informática
Escuela de Ingeniería y Arquitectura
Universidad de Zaragoza

Diciembre 2014

AGRADECIMIENTOS

A Ángel, por toda su paciencia y apoyo. No hubiera sido posible de otro modo.

A Sergio y Óscar por su orientación y ayuda.

A mi familia: Jorge, Pilar y Alberto, por compartir el tiempo que les correspondía a ellos.

A mi madre.

RESUMEN

El generador automático de fichas de personajes pretende hacer una pequeña aportación en la gestión y aprovechamiento de la información. En este caso, este PFC se centra en el mundo del periodismo y la gestión documental, donde un artículo, una fotografía, y en general, cada elemento que aparece en un periódico, supone un valor añadido que aporta información, y sirve como base para escribir nuevos artículos, ahorrando así tiempo y dinero. Cuando un documentalista en un periódico realiza una búsqueda habitual, los resultados son una lista de enlaces a páginas, idéntica a la que recibe cualquier usuario con un buscador web. Esta herramienta lo que permitirá es filtrarlos, organizarlos, estructurarlos y presentarlos de una forma adecuada, lo cual supondrá un salto cualitativo en la recuperación de la información.

El objetivo principal de este proyecto es desarrollar una aplicación que permita generar de forma automática fichas informativas estructuradas, centrado en el caso concreto de los personajes que aparecen en las noticias. Se parte de los textos y fotos pertenecientes a una base de datos documental de un medio de comunicación, pero también utilizaremos otras fuentes de Internet, incluyendo bases de datos construidas con *linked data* y redes sociales.

Este PFC se desarrolla en Ibercentro Media Consulting & Services S.L., que es una empresa que presta servicio informático al Grupo Heraldo y Diario de Navarra.

Las contribuciones fundamentales de este proyecto son:

1) Realizar un estudio del estado del arte a partir de artículos de investigación en tres áreas:

- El procesamiento del lenguaje natural y la extracción de datos, la desambiguación de nombres, y la búsqueda en una base de datos documental. Esto es necesario porque un importante objetivo es el reconocimiento de patrones para localizar determinados datos que se encuentran embebidos en textos o en páginas web.

- El proyecto de Web Semántica y el uso de *linked data* en la web, como otro modo de localizar información en Internet.

- La identificación de imágenes de primer plano, para completar la información sobre las imágenes almacenadas en el periódico.

2) Desarrollar un prototipo funcional que formará parte de la plataforma de documentación, enmarcado en concreto entre los procedimientos de búsqueda de información de que actualmente funcionan sobre la plataforma documental de contenidos de Heraldo de Aragón y Diario de Navarra.

Finalmente añadir que este trabajo ha dado fruto a un artículo de investigación donde he participado como co-autora.

ÍNDICE

MEMORIA PRINCIPAL

1. INTRODUCCIÓN

En esta sección se describirán a un nivel muy general los aspectos más importantes del presente proyecto de fin de carrera, en adelante PFC, con el objetivo de proporcionar al lector una visión global del trabajo realizado.

Al realizar una búsqueda en una gran cantidad de información, como sucede en la WWW o en este caso en el archivo de un periódico, obtenemos una gran cantidad de resultados. Esto nos lleva a buscar soluciones para localizar información más rápido y de forma más precisa. En particular, la gestión documental en la empresa demanda herramientas que adopten estas soluciones y proporcionen al entorno empresarial un control de costes en tiempo y dinero. En concreto en un periódico, cuando un periodista solicita información a un documentalista, éste tiene que buscar de forma manual en el archivo documental y revisar los resultados para localizar lo que se le solicita. Además, si quiere buscar información en otras fuentes, es un coste de tiempo añadido.

Este proyecto aporta en el contexto de la empresa el estudio y desarrollo de un generador de fichas biográficas que facilita la obtención de toda la información disponible sobre un personaje (por ejemplo, un político, un futbolista o una cantante) extrayéndola tanto de los recursos propios (la base de datos de archivo de un medio escrito) como de Internet. Para abordar este proyecto, ha sido necesario estudiar métodos de acceso a repositorios semánticos y herramientas de última generación para el procesamiento de lenguaje natural (PLN) y minería de textos.

1.1 Contexto

Este PFC se ha realizado en la empresa Ibercentro Media Consulting & Services S.L., concretamente en el Área de Innovación Tecnológica, bajo la dirección de Ángel Luis Garrido Marin, co-dirección de Óscar Gómez Rubio, y en paralelo, como ponente, en la Universidad de Zaragoza el profesor Sergio Ilarri.

Ibercentro Media Consulting & Services da soporte a los medios “Heraldo de Aragón”, “Diario de Navarra” y “Heraldo de Soria”. El trabajo realizado interactúa con una plataforma desarrollada por la empresa para la gestión documental llamada EMMA[1] que a su vez se relaciona con otras aplicaciones en las que se produce el periódico. Ésta se explica más detalladamente en el Anexo B, apartado “Contexto”.

1 EMMA: http://www.hiberus.com/gestion-de-archivos-procesos-de-documentacion-gestor-documental-emma

1.2 Motivación

Un periodista es la persona que se dedica profesionalmente al periodismo, en cualquiera de sus formas, ya sea en la prensa escrita, radio, televisión o medios digitales. Su trabajo consiste en descubrir e investigar temas de interés público, contrastarlos, sintetizarlos, jerarquizarlos y publicarlos. Para ello recurre a fuentes periodísticas fiables y verificables. Así elabora sus artículos, que pueden tomar varias formas para su difusión: oral, escrita, visual. En la elaboración cuenta con el apoyo inestimable de los documentalistas, profesionales que están formados para gestionar la información dentro de las organizaciones.

La labor del documentalista es compleja, debido a las grandes cantidades de información que manejan, y de las que tienen que filtrar los datos que se les requiere por parte de los periodistas.

La idea de elaborar este generador automático de fichas biográficas surge de la necesidad de los documentalistas a la hora de transmitir datos a los periodistas sobre un determinado personaje. La búsqueda entre los artículos y fotografías archivados conlleva una gran cantidad de tiempo y esfuerzo para un personal que tiene muchas otras tareas que llevar a cabo.

Para realizar esta herramienta, se han consultado posibles soluciones ya existentes. En páginas web se encuentran algunas como:

http://omnibiography.com/ - Genera una biografía pero hay que proporcionarle todos los contenidos. Es únicamente una plantilla.

http://www.vizify.com/ - Vizify es una aplicación online que permite generar una biografía basada en Social Media, más precisamente en redes sociales específicas como Facebook, Twitter, Foursquare y Linkedin. Esta información es almacenada, reordenada y mostrada como un Mapa Mental, que se convierte en una especie de cuadro sinóptico de nosotros mismos.

http://www.biographyonline.net/ Tiene fichas prediseñadas con biografía, fotos y enlaces a páginas Web relacionadas. No puedes elegir el personaje sino sólo escoger uno de los que te ofrece el índice.

Respecto a aplicaciones que directamente utilicen la información almacenada en el archivo de un periódico, no se ha encontrado ninguna, por lo que se puede destacar la novedad de este trabajo.

El generador de fichas biográficas permitirá a los usuarios obtener directamente la información más relevante sobre un personaje concreto a partir de toda la que hay almacenada en la base de datos del periódico, y además completarla con otros datos extraídos de Internet.

1.3 Dificultades

Vistas las necesidades descritas anteriormente y puestos a encontrar una solución, se encuentran algunas dificultades iniciales que requieren una atención especial:

- Extraer una información determinada de una gran cantidad almacenada en forma de textos y fotos en el archivo documental. Buscamos los datos e imágenes que necesita el usuario sobre un personaje en concreto. Para ilustrar estas cantidades se muestran a continuación unas estadísticas del crecimiento de toda esta información hasta 2013, y que en 2014 continúa en aumento (Tabla 1.1):

AÑO	1994	1995	1996	1997	1998	1999	2000
FOTOS	177	3254	32183	44956	48660	68959	100909
PAGINAS	32064	31600	31756	32137	33024	33744	29542
TEXTOS	63977	66675	66527	67071	68173	69201	69982

AÑO	2004	2005	2006	2007	2008	2009	2010	2011	2012	2013
FOTOS	190518	231967	202482	175998	174972	155625	136933	115204	104958	115178
PAGINAS	42694	82336	72338	66113	66181	57057	56004	55609	53270	51664
TEXTOS	95350	96709	83484	108301	105634	117736	126064	130578	126940	116725

Tabla 1.1: Cantidades almacenadas cada año en el banco de contenidos

A continuación se muestra una estadística de cómo ha evolucionado el almacenamiento de fotografías (fig. 1.1) y archivos PDF (cada archivo contiene una página de un ejemplar de periódico. Fig. 1.2).

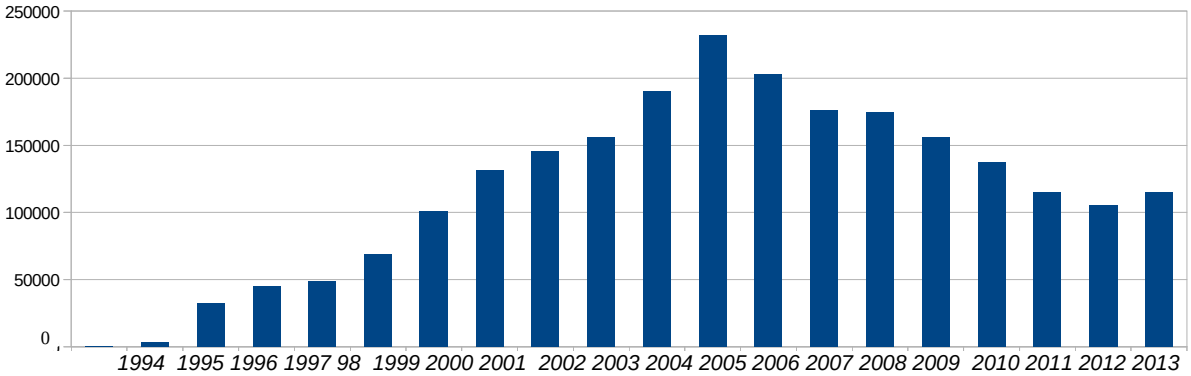

Figura 1.1: Evolución del número de fotos almacenadas

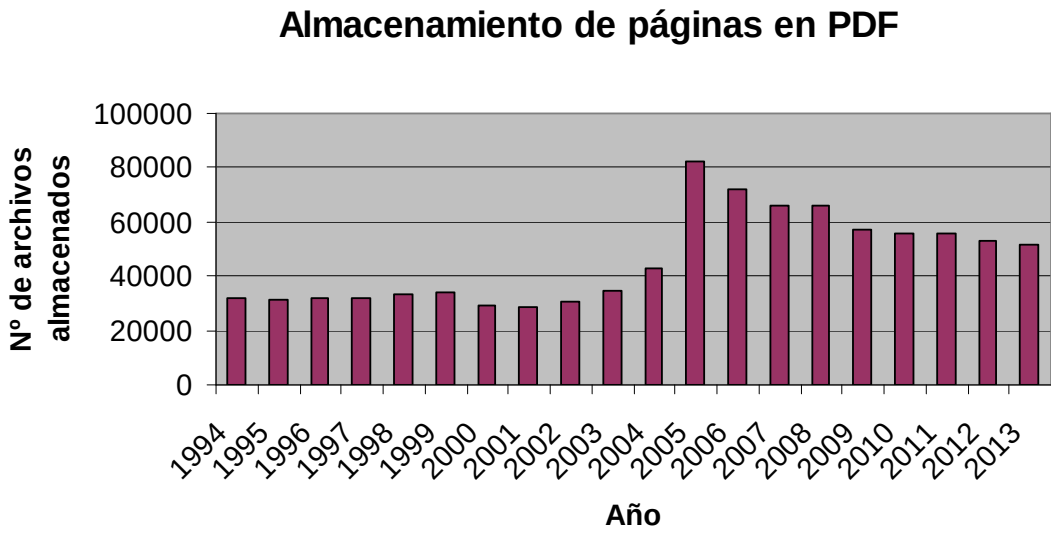

Figura 1.2: Evolución del número de páginas almacenadas en formato PDF

- Otra dificultad surge en la forma de almacenar las fotografías. Aunque el sistema contempla la etiqueta que recoge el tipo de plano para cada una de ellas, la mayoría de estas etiquetas están en blanco. Por tanto, si queremos añadir una foto del personaje tenemos que determinar si las fotografías que encontremos son o no primeros planos, a partir de algoritmos implementados en esta nueva herramienta que lo hagan posible. Actualmente hay alrededor de 2.700.000 fotos almacenadas.

- El sistema debe resolver los conflictos de ambigüedad entre dos personajes con un mismo nombre o entre un personaje y otro nombre propio. Por ejemplo, si buscamos información sobre (Amaia) Salamanca y el usuario no nos proporciona el nombre de pila, eliminar toda la información referida a Salamanca como ciudad u otras personas apellidadas Salamanca. La resolución de ambigüedad entre personajes, nos lleva además a la creación previa de un catálogo de *Named Entities* (Entidades con nombre, en nuestro caso, personajes que aparecen en el periódico. Ver en memoria principal, capítulo 3.1).

- Y finalmente, la última dificultad que se destaca en este capítulo es el extraer información de un repositorio como DBpedia (ver capítulo 2.8 de esta memoria). DBpedia es un proyecto para la extracción de datos de Wikipedia y transformarlos en un repositorio semántico. El proyecto DBpedia está realizado por la Universidad de Leipzig, Universidad Libre de Berlín y la compañía OpenLink Software.

1.4 Objetivos

Los objetivos de este PFC son los siguientes:

- Investigar y tomar contacto con las técnicas de procesamiento del lenguaje natural, para encontrar modos de extraer la información que necesitamos al generar biografías a partir de los artículos periodísticos.

- Resolver los conflictos de ambigüedad entre nombres propios de manera óptima estudiando los posibles algoritmos y técnicas para desambiguar.

- Buscar formas de identificar fotografías de primer plano, ya que el sistema de almacenamiento existente no lo hace.

- Tomar contacto con el concepto de Web Semántica, y utilizarlo para extraer información de páginas web que utilicen *linked data*, mediante consultas SPARQL.

- Utilizar servicios web o librerías que ofrecen los buscadores en Internet para encontrar información sobre un personaje: biografías, fotos, vídeos, redes sociales.

- Desarrollar e implantar en el área de documentación un prototipo que permita la obtención de una ficha biográfica sobre un determinado personaje, aprovechando las informaciones existentes en los artículos y en las fotos almacenadas, y añadiendo información extraída de Internet, incluyendo la que está en forma de *linked data.*

1.5 Tecnologías utilizadas

Las principales tecnologías y herramientas utilizadas en este proyecto son:

En cuanto a lenguajes de programación, se ha utilizado normalmente *la plataforma Visual Basic .Net (VB.NET) [3].*Por otra parte, se ha utilizado un entorno de desarrollo como *Visual Studio 2010 [4]* para el lenguaje de programación indicado, y SQL Server [6] como gestor para Bases de Datos. Estos son los estándares de trabajo impuestos por la empresa.

Como software de procesamiento de lenguaje natural (PLN) para tratamiento de textos, se ha utilizado *Freeling [10]*, conjunto de librerías abiertas (*open source*) desarrolladas por la Universidad Politécnica de Cataluña. Está codificado en C++, funciona en Linux y Windows y tiene una API de llamadas para distintos lenguajes, entre ellos Java que fue el que se utilizó en el primer prototipo. Sin embargo, ya que el sistema operativo que se utiliza en el medio periodístico donde se ha desarrollado este PFC es Windows, finalmente se ve más adecuada la versión de Freeling para Windows, de posterior aparición.

Por otro lado se han utilizado varios estándares entre los que cabe destacar:

- **SQL** [11] es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones en ellas.
- **XML** [12] eXtensible Markup Language ('lenguaje de marcas extensible'), es un lenguaje de marcas desarrollado por el World Wide Web Consortium (W3C). Deriva del lenguaje SGML y permite definir la gramática de lenguajes específicos (de la misma manera que HTML es a su vez un lenguaje definido por SGML) para estructurar documentos grandes.
- **RDF** [13] el Marco de Descripción de Recursos (del inglés Resource Description Framework, RDF) es un framework para metadatos en la World Wide Web (WWW), desarrollado por el World Wide Web Consortium (W3C). Es un lenguaje de propósito general para representar información en la web.
- **SPARQL** [14] SPARQL es un acrónimo recursivo del inglés SPARQL Protocol and RDF Query Language. Se trata de un lenguaje estandarizado para la

consulta de grafos RDF. Es una tecnología clave en el desarrollo de la Web Semántica, que se constituyó como Recomendación oficial del W3C.

- **HTML** [18], siglas de HyperText Markup Language («lenguaje de marcas de hipertexto»), hace referencia al lenguaje de marcado para la elaboración de páginas web. Es un estándar que sirve de referencia para la elaboración de páginas web en sus diferentes versiones, define una estructura básica y un código (denominado código HTML) para la definición de contenido de una página web, como texto, imágenes, entre otros. Es un estándar a cargo de la W3C, organización dedicada a la estandarización de casi todas las tecnologías ligadas a la web, sobre todo en lo referente a su escritura e interpretación.

En general se ha utilizado la plataforma *Windows* tanto para el desarrollo de aplicaciones como para la ejecución de programas de usuario ya que es la única plataforma que se usa en la empresa. También fue necesario instalar el sistema operativo Linux (en concreto, *Ubuntu)* para poder usarlo sin problemas en el prototipo como un servicio web en Java. Como se ha dicho antes, las últimas versiones ya funcionan en entorno Windows.

Por último, respecto al tipo de desarrollos realizados encontramos aplicaciones de escritorio, servicios de Windows [16], y librerías de clases (DLL en .NET y JAR en Java) [17] y servicios Web [15].

1.6 Fases del proyecto

La primera etapa del proyecto sin duda ha sido la investigación sobre técnicas de procesamiento del lenguaje natural (PLN) y el análisis de diferentes algoritmos que permiten la identificación de elementos concretos dentro de un texto. También sobre otras técnicas de minería de textos que permitan localizar patrones embebidos en textos y que nos proporcionen otros datos requeridos. En segundo lugar, el estudio del estado del arte respecto a la Web Semántica y su uso práctico, para entender y utilizar como fuente de datos a DBpedia. Y en tercer lugar, el estudio de una aproximación a técnicas de reconocimiento facial, con el fin de identificar fotos de primer plano. Finalmente, ha sido necesario conocer el contexto de trabajo (plataforma documental EMMA) y los flujos de trabajo. Los estudios realizados y artículos estudiados pueden verse con más detalle en el Anexo A de esta memoria.

Tras todo el proceso de investigación, podemos distinguir en la elaboración del proyecto las siguientes fases:

1 – **Análisis de requisitos del sistema**. Interesante dentro del análisis, la necesidad de crear un catálogo de *Named Entities* a partir de la información almacenada en el archivo documental, que será necesario para el funcionamiento del generador de biografías. Aquí utilizaremos técnicas de minería de textos analizadas en los estudios previos, que pueden consultarse en el Anexo A de la memoria.

2 – **Desarrollo de un prototipo**. Este primer prototipo incluye las mismas funcionalidades excepto la forma de presentarse al usuario y una búsqueda en Internet más limitada que no incluye *Linked Data*.

3 – **Diseño de la solución**. Dentro del diseño, podemos destacar la necesidad de escoger un algoritmo para desambiguar *Named Entities.* Valorando los resultados de la aplicación de los algoritmos estudiados (Anexo A), y de las posibilidades prácticas de aplicarlos, optaremos por una implementación utilizando una desambiguación basada en un catálogo previo de *Named Entities*, y en el tesauro[2] del archivo documental. También nos apoyaremos en la búsqueda dentro de un repositorio semántico como algo novedoso y que nos puede aportar datos que faciliten la tarea.

4 – **Desarrollo del generador de fichas biográficas**. La codificación propiamente dicha, donde se puede destacar como mayor dificultad la creación de módulo que realice las consultas SPARQL necesarias para extraer información del repositorio semántico que es DBpedia, la creación de un catálogo propio de Named Entities, o la desambiguación entre Named Entities.

5 – **Pruebas**. Donde la importancia de comprobar los tiempos de acceso ha sido imprescindible para conseguir una herramienta que realmente sea útil. Tanto esta fase como la anterior están documentadas en el Anexo B.

6 – **Puesta en marcha**. La integración dentro del conjunto de herramientas utilizadas en la empresa, que se explica con más detalle en el Anexo B.

1.7 Organización de la memoria

Este primer capítulo expone el contexto en el que se realizó el proyecto, además de la definición de objetivos y fases marcados para la realización del proyecto y las tecnologías utilizadas. además del desarrollo de las fases del proyecto.

El segundo capítulo describe el trabajo previo a la implementación de la herramienta, incluyendo la planificación, la realización de un prototipo, y el estudio de soluciones para tres dificultades concretas que aparecen al analizar el análisis: la creación de un catálogo de personajes, la identificación de fotografías de primer plano, y la valoración de la importancia de una noticia. También incluye la planificación, tiempos estimados y reales de todo el proceso de realización del proyecto.

En el capítulo 3 se incide en la arquitectura en la que se basa el proyecto de un modo muy general. Toda la información se completa y amplía en el Anexo B de la memoria.

El cuarto capítulo contiene la metodología de trabajo y los resultados, reflejados en encuestas de satisfacción de los usuarios.

Y para terminar, en el capítulo cinco se enmarcan las conclusiones y valoración personal del trabajo llevado a cabo. A su vez, se abordan las posibles líneas de trabajo futuro en el campo de la generación de biografías.

La memoria principal va seguida de tres anexos:

[2] Un tesauro es una lista de palabras o descriptores que se usan para representar conceptos (por ejemplo, personas).

- El primero (Anexo A) contiene los estudios realizados sobre el estado del arte en relación al procesamiento del lenguaje natural, al reconocimiento facial y la Web Semántica.

- El Anexo B detalla en más profundidad todo el desarrollo de este PFC. En él se incluyen el análisis y diseño completos, y las fases de desarrollo y pruebas.

- En el Anexo C aparece el borrador del artículo de investigación realizado: *"Generating automatic data sheets from mixed environments – Experience in a Media Company".*

2. PREPARACIÓN DEL CONTEXTO DE TRABAJO

En este capítulo abordaremos una visión general de la herramienta, para ofrecer una descripción que permita una comprensión global de su funcionamiento, que luego se ampliará en detalle en el Anexo B.

También se describen los trabajos previos a la realización del generador de fichas biográficas, y a raíz de los resultados obtenidos con el prototipo sobre tiempos de ejecución. Estos últimos han llevado al desarrollo adicional en este PFC de tres pequeñas herramientas que realizan las siguientes tareas:

- Para poder ejecutar la búsqueda previa del personaje, era necesario añadir a la base de datos del periódico una tabla que enlace las *Named Entities* de personajes con las etiquetas usadas EMMA para localizar a estos personajes en noticias o fotografías. Para ello se creó el catálogo de *Named Entities* que se explica en la sección 2.4.

-Las fotos de primer plano que se van a mostrar en la ficha biográfica no están etiquetadas como tales en la base de datos de EMMA. Como analizar cada foto al realizar la búsqueda llevaría demasiado tiempo, se ha desarrollado una herramienta de etiquetado de fotografías que también se describe en la sección 2.5.

-La restricción de tiempos de ejecución también lleva a decidir el etiquetado previo de las noticias, asignando una puntuación a cada una. Este proceso se explica en la sección 2.6. De esta manera, el generador de fichas biográficas accede directamente al “ranking” de noticias más relevantes para cada personaje.

Todas estas soluciones surgen del estudio previo de artículos de investigación que se puede consultar en el Anexo A de esta memoria.

2.1. Objetivos y resumen de la herramienta

El objetivo principal de este proyecto es desarrollar un sistema que permita generar la ficha informativa de un personaje de forma automática a partir de las noticias pertenecientes a una base de datos documental de un medio de comunicación y de datos que podamos extraer de Internet. La aplicación informática se encarga de generar documentos electrónicos que contengan información relacionada con el personaje seleccionado que consiste en una selección de textos y fotografías relevantes por su actualidad o por la importancia de la noticia, datos biográficos, enlaces a páginas, vídeos, redes sociales u otros enlaces relevantes a información que aparezca en Internet.

En la figura 1.3 se esquematiza el proceso que sigue la información en el medio periodístico: los documentos se generan a través de una aplicación externa llamada Milenium [1], que los depositará en el archivo documental. Nuestra herramienta accede tanto a ellos como a información en la Web para obtener la ficha informativa.

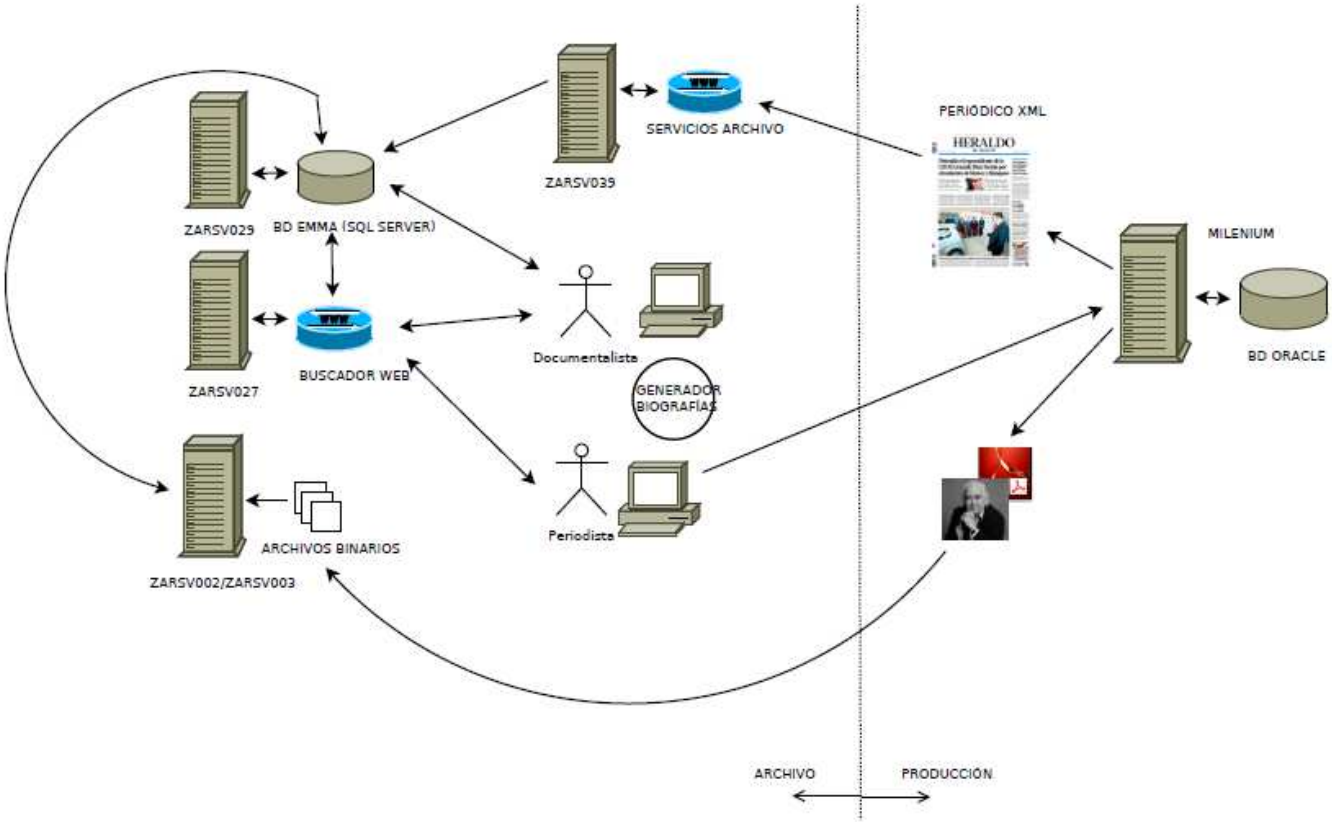

Figura 1.3: Situación del Generador de Biografías dentro del proceso de archivo y recuperación de información del Heraldo de Aragón.

En la figura 1.3 se pueden ver los diferentes servidores donde se almacena la información: ZARSV002 y ZARSV003 almacenan las fotografías y los archivos PDF que contienen cada página de cada periódico publicado. El servidor ZARSV029 contiene la base de datos EMMA a la que se accede para extraer textos de artículos publicados, y el servidor ZARSV027 contiene el indexador para agilizar las búsquedas en la base de datos. Los usuarios finales, documentalistas y periodistas tienen instalado en sus equipos el prototipo de este PFC para generación de fichas de personajes. El generador de fichas biográficas pasará a formar parte de la plataforma EMMA que ya se ha mencionado en el capítulo 1 de esta memoria.

2.2 Planificación

En este apartado se resumen tanto los tiempos planificados como los invertidos, para dar una idea del trabajo realizado y de la distribución del mismo en las distintas etapas.

La planificación del proyecto, descompuesta en las actividades principales es (la barra superior indica la duración de las tareas en horas):

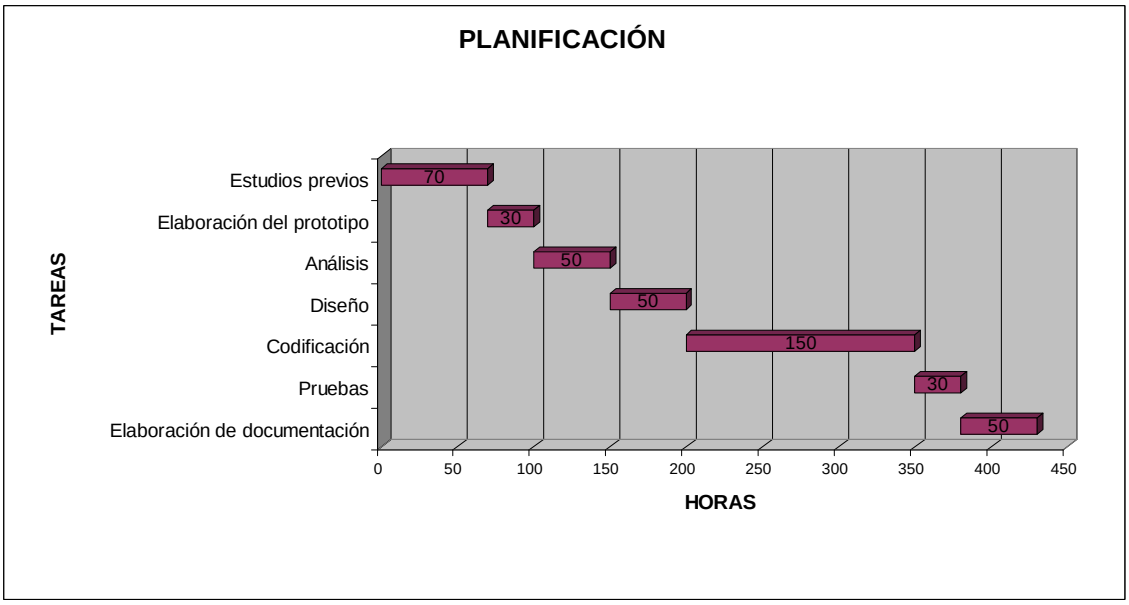

Fig. 1.4: Diagrama de Gantt de planificación

En el diagrama de Gantt podemos ver una suma total acumulada de 430 horas de trabajo estimadas.

Respecto a las horas reales, han sido más de las previstas, excepto las dedicadas al análisis y diseño, y a la elaboración de la documentación. En particular, las dedicadas a la codificación han sido sensiblemente mayores debido al desfase en la formación personal, y la necesidad de puesta al día en estándares y tecnologías actuales.

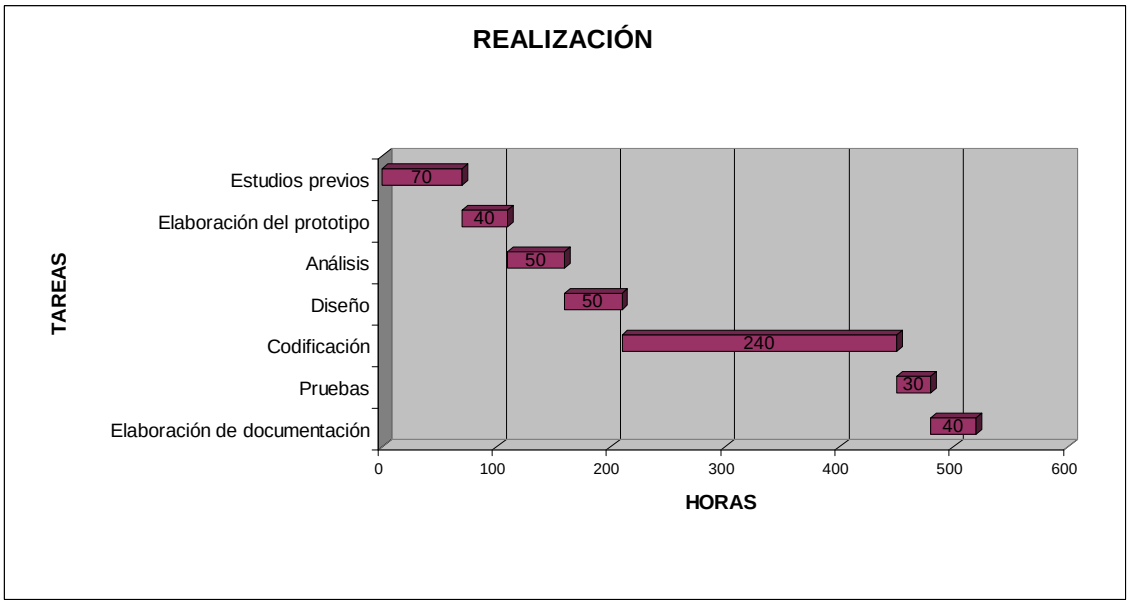

Fig. 1.5: Diagrama del desarrollo

En la tabla que se ve a continuación (Tabla 1.2) se resumen las horas empleadas para completar cada una de las tareas del proyecto:

Tarea	Planificación	Duración real
Estudios previos	70 horas	70 horas
Elaboración del prototipo	30 horas	40 horas
Análisis	50 horas	50 horas
Diseño	50 horas	50 horas
Codificación	150 horas	240 horas
Pruebas	30 horas	30 horas
Elaboración de documentación	50 horas	40 horas
TOTAL	**430 horas**	**520 horas**

Tabla 1.2: Resumen temporal

2.3. Prototipo inicial

Tras un primer análisis, se realizó un prototipo con unas funcionalidades básicas: se limitaba a la búsqueda sobre todo de noticias y fotografías que eran más recientes y más relevantes. Esto permitió analizar el tiempo de acceso a la base de datos de archivo. También se incluyó la obtención de algunos enlaces en Internet, únicamente los primeros resultados que proporcionaba (ahora ya está obsoleto) el Servicio web de Bing. En la figura 1.6 podemos ver la pantalla inicial para el uso de documentalistas y periodistas, donde se recoge el nombre el personaje del que se desea obtener la ficha biográfica.

Figura 1.6: Pantalla inicial para búsqueda rápida

El resultado del prototipo se obtiene en un archivo XML, del que vemos un fragmento en la figura 1.8.

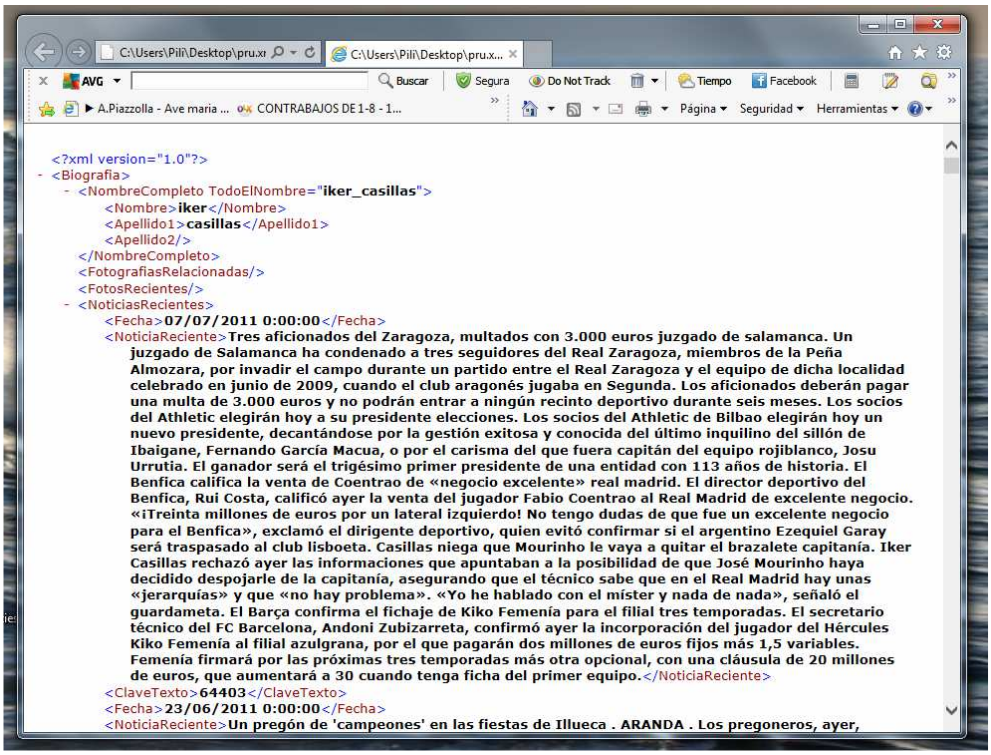

Figura 1.7: Archivo XML resultante

Este prototipo fue usado para una primera batería de pruebas, realizadas por personas conocedoras de los requisitos y el contexto, con experiencia en desarrollo. De esta batería surgieron errores y sugerencias que nos permitieron afinar el funcionamiento del sistema y, sobre todo, tener en cuenta los tiempos necesarios de búsqueda que nos limitan los repositorios de información y otros medios utilizados.

Estos tiempos de búsqueda han sido muy importantes a la hora de tomar decisiones de diseño, ya que el formato de las consultas SQL a la base de datos, sobre todo a la hora de escoger las noticias más relevantes, podían llevarnos a la necesidad de crear una herramienta que hiciera la búsqueda en tiempo real, o en cambio a plantearnos crear un repositorio previo de información sobre los personajes más relevantes.

Finalmente se tomó una decisión intermedia. Dado que los cuellos de botella eran la recuperación de noticias relevantes (implica recorrer todas las noticias del personaje, tarda entre 2 y 3 minutos en los equipos descritos anteriormente) y encontrar las fotografías de primer plano (no estaban etiquetadas, por lo que había que filtrar también todas las fotos de ese personaje), se ha realizado un etiquetado para cada noticia y fotografía que permite realizar la ficha biográfica en tiempo real.

Estas soluciones están desarrolladas en las secciones que siguen a continuación.

2.4 Creación de un catálogo de *Named Entities*

Este catálogo se utiliza más adelante en el generador de biografías para ayudarnos en la búsqueda previa, en la desambiguación de nombres y para la búsqueda de textos y fotos (ver Anexo A, capítulos 2 y 3). Para su elaboración se han dado los siguientes pasos (Fig. 1.11).

Proceso:

1. Extraer de DBpedia la lista de todos los personajes. De todas sus propiedades elegir “name”.

2. Recorrer el XML obtenido de DBpedia. Para cada personaje:
- Extraer el nombre completo (valor de la propiedad)
- Almacenarlo

3. Extraer la lista de personajes del tesauro del medio periodístico.

4. Crear la tabla CATALOGO_NE en la que se enlazan las dos listas anteriores:

- Para cada elemento del tesauro de textos identificado como PERSONAS,
  - ✓ Buscar que el nombre del personaje esté dentro de la lista extraída de DBpedia. Si está, crear un registro con ese personaje de esa lista y el campo id del tesauro.
  - ✓ Si no está, crear una entrada en el catálogo con el nombre e id del tesauro.

- Para cada elemento del tesauro de fotos identificado como PERSONAS,

  - ✓ Buscar que el personaje que contiene esté dentro de la lista de DBpedia. Si está en la lista:
    - buscar si ese personaje ya está en la tabla y añadir al registro el campo clave del tesauro.
    - Si no está en la tabla, crear el registro con ese personaje de la lista y el campo clave del tesauro.
  - ✓ Si no está en la lista de DBpedia, crear una entrada en el catálogo con el nombre e identificador de la entrada en el tesauro.

5. Completamos el catálogo con personajes que aparecen en los textos de los artículos y que no están ni en DBpedia ni en el tesauro.

Para cada texto almacenado correspondiente a una noticia:
- Ejecutar Freeling [10] sobre el texto:
  - ✓ Extraer *Named Entities(NE)* de él y almacenarlas
  - ✓ Almacenar lematizado verbo anterior y posterior (si existe)
- Con el *Gazetteer* [11] que forma parte del entorno de EMMA, detectamos de un artículo las NE correspondientes a lugares geográficos y las eliminamos de la lista anterior.

- Con una lista de organizaciones, empresas y partidos políticos extraída de Internet, comparamos y eliminamos estas entidades de la lista anterior.

- Para cada una de las NE restantes:
  - ✓ Actualizamos el contador de cada NE por número de apariciones, y se suma a la puntuación anterior.
  - ✓ Eliminamos las NE con la puntuación por debajo de un umbral determinado.

6. Para cada uno de los personajes extraídos del punto anterior:
   - Buscarlo en el catálogo de *Named Entities*.
   - Si no está, almacenarlo como una nueva entrada.

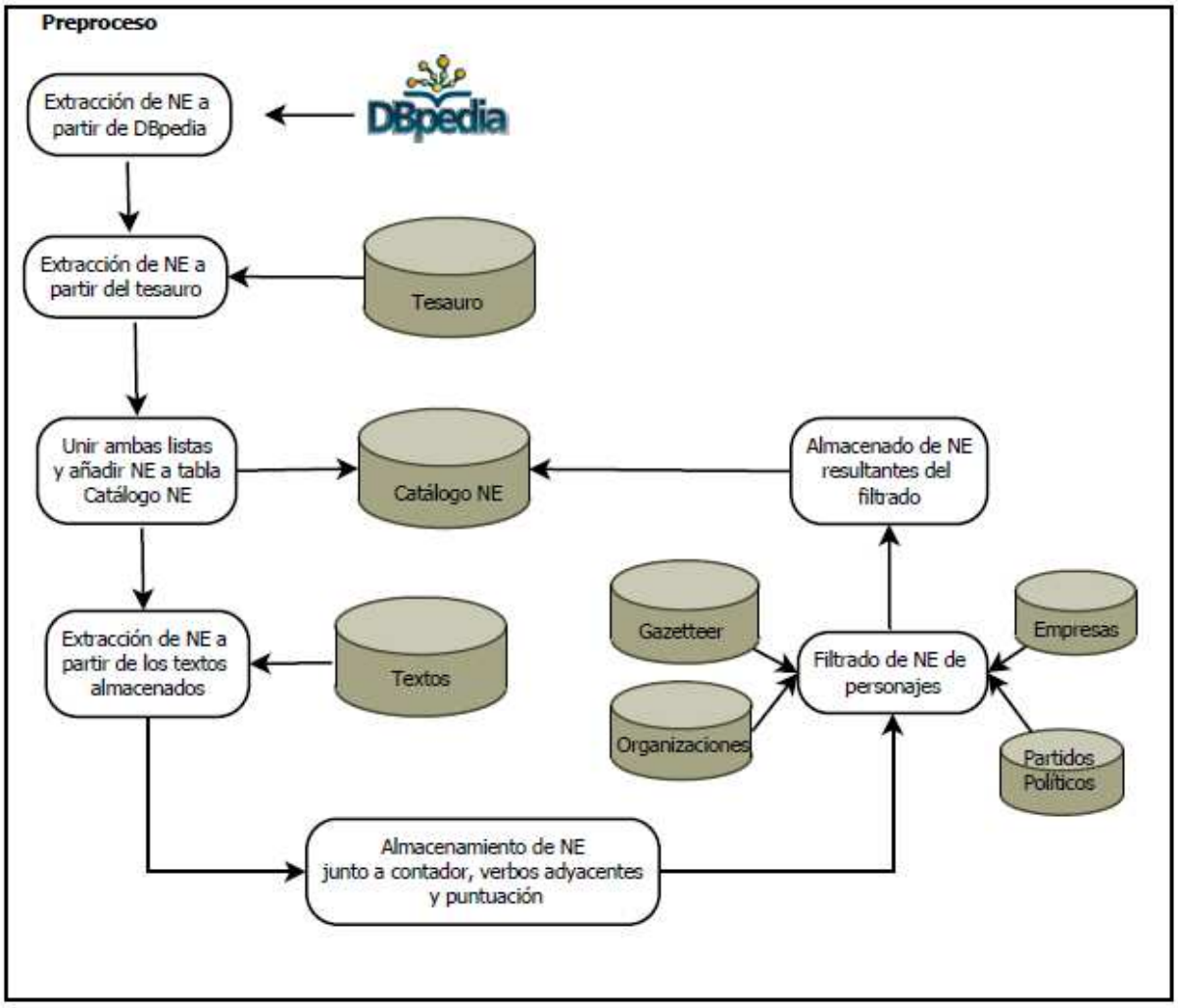

Figura 1.8: Diagrama de actividades ampliado del preproceso

El algoritmo se completa usando un *Gazetteer* y listas de otras entidades, porque la herramienta *Freeling*, aunque posee un módulo de etiquetado (Module BIONER)[3] que teóricamente detecta NE de personas, en la práctica y según pruebas realizadas en el Heraldo con anterioridad, este etiquetado no es fiable. La herramienta tipo gazeteer [GABUILME13], es la usada en Heraldo de Aragón para almacenar localidades. Dicho gazetteer está construido a partir de la base de datos geográfica Geonames [4].

[3] http://nlp.lsi.upc.edu/freeling/doc/userman/html/node40.html
[4] http://www.geonames.org/

Este procesado de datos se realiza de forma completa sobre todo el conjunto de información tal como se indica en el esquema de la Figura 1.8 la primera vez que se crea el catálogo. Después, con una periodicidad bimensual, este proceso se volverá a ejecutar de forma incremental, añadiendo los resultados nuevos que van apareciendo. Este proceso será supervisado por el personal de documentación.

Finalmente, se ha incorporado el catálogo como una tabla a la base de datos del archivo documental. Y mientras se almacena cada personaje, éste se relaciona con sus identificadores en el tesauro de textos e imágenes de la base de datos EMMA y con su enlace a su URI[5] correspondiente en la DBpedia.
De esta manera, el catálogo nos ha permitido unificar la denominación de los personajes y a la vez localizarlos con mayor exactitud dentro (intranet) y fuera (Internet) del entorno documental en el que se trabaja.

En la práctica, debido a limitaciones temporales, el catálogo está realizado únicamente a partir de los personajes extraídos de DBpedia y del tesauro del periódico. En total, tiene una extensión de 10.500 personajes.

2.5 Etiquetado de fotografías de primer plano

Después de realizar el estudio del arte sobre este tema (ver Anexo A, sección 2), se llega a la conclusión que las técnicas de desarrollo facial han pasado de ser uno de los grandes retos a ser parte habitual de herramientas de uso frecuente, como puede ser Facebook en su etiquetado de fotografías.

En el archivo documental del Heraldo de Aragón hay a fecha de hoy, un total de 2.700.000 fotografías almacenadas. De ellas sólo 70.000 tienen etiqueta con el tipo de plano que contienen. Lo mismo nos sucede con el personaje al que muestran: en el campo correspondiente al tesauro para fotografías, sólo unas 200.000 etiquetan a un personaje concreto.

Si el generador de fichas biográficas trata de determinar el personaje y el tipo de plano de cada fotografía obtenida en una consulta en tiempo real, el tiempo de ejecución sube excesivamente Por ejemplo, para el personaje “Jose Luis Rodriguez Zapatero”, de frecuente aparición en los periódicos, sólo en el año 2010 tenemos 218 fotografías que contienen en su descripción ese nombre, y tan sólo 20 que tengan la etiqueta de esa persona en el tesauro. Para un solo año tenemos casi 240 fotografías que analizar con el software de reconocimiento facial para determinar si son primeros planos o no, y el tiempo total sólo para localizar fotografías nos lleva más de 15 minutos.

Por tanto, se nos plantea la necesidad de realizar un etiquetado previo de las fotografías, tanto para los personajes que muestran, como del tipo de plano que contienen.

Ambos etiquetados se realizan con la herramienta de etiquetado de fotografías que se describe con más detalle en el Anexo B, sección 3.3.2, y se muestra a continuación en la figura 1.9. En esa sección también pueden consultarse los resultados que arrojan.

Para el etiquetado de personajes, para cada personaje del catálogo de *Named Entities* descrito en el capítulo anterior, se recorren todas las fotografías buscando en el

[5] Identificador de recursos uniforme o URI —del inglés *Uniform Resource Identifier*— es una cadena de caracteres que identifica los recursos de una red de forma unívoca, en este caso obtenida de DBpedia.

tesauro y la descripción. Si están publicadas, se mira también en el campo del texto de la noticia correspondiente que almacena el pie de página.

Si aparece el personaje, se crea una referencia que enlace la fotografía con el catálogo de *Named Entities* y se actualiza el campo del tesauro de la fotografía.

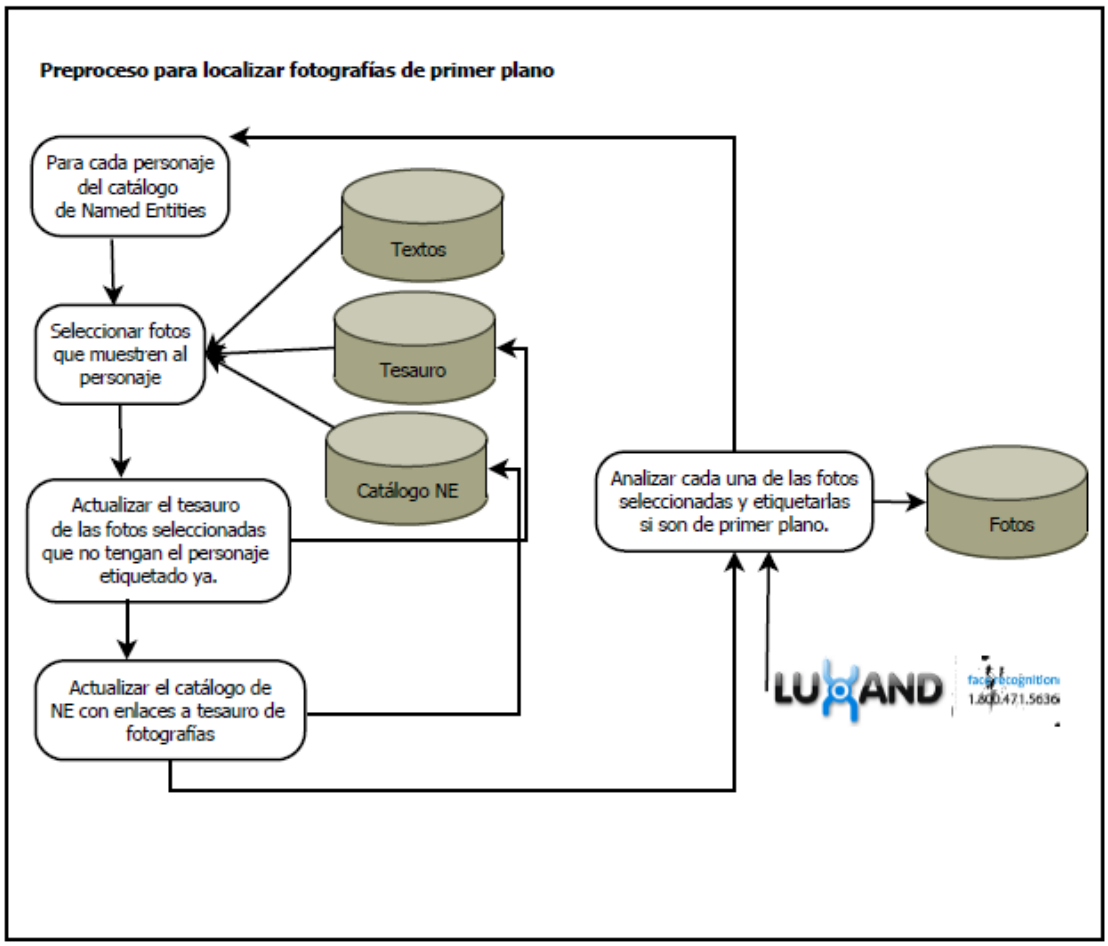

Fig. 1.9: Diagrama de actividades del preproceso para fotografías de primer plano.

Para el segundo problema, que es concluir si una foto es de primer plano o no, se han utilizado unas librerías desarrolladas por la empresa Luxand[6], denominadas FaceSDK 5.0. Éstas permiten la detección de caras, devolviendo el tamaño de la imagen detectada. El porcentaje de la cara sobre el área total de la fotografía nos permite determinar si es un primer plano o no. Por tanto, se analiza con esta herramienta cada fotografía y la etiquetamos con el tipo de plano obtenido.

Estas librerías de reconocimiento facial se han escogido por haber sido utilizadas ya previamente en el Heraldo.

[6] http://www.luxand.com/facesdk/

2.6. Elaboración de un "ranking" de noticias más relevantes por personaje

Para poder elegir las noticias más relevantes, hemos tenido también que diseñar un algoritmo que nos dé los resultados más cercanos a los deseados. Para ello se ha utilizado información extraída de diferentes artículos de investigación, de los que destaca el propuesto por Luhn en 1958 [LUHN58], y del que sacamos la idea (algoritmo de ponderación basada en frecuencias de aparición de palabras y prefiltrado) para destacar una noticia sobre otras. Se basa en que las palabras que más se repiten, son más relevantes, excepto categorías léxicas cerradas (determinantes, pronombres, preposiciones, verbos auxiliares…). En este caso, se utiliza la aparición del nombre del personaje.

En primer lugar (ver Fig. 1.10), se cuenta el número de palabras de la noticia, seleccionando las que superen un determinado umbral. Se puntúan las apariciones del nombre del personaje tanto en el título de la noticia, el pie de foto (si lo tiene), como en el texto, y en los campos *keywords* y *resumen.* También se valorará si el personaje aparece en la portada, en la contraportada, en página impar o en un monográfico.

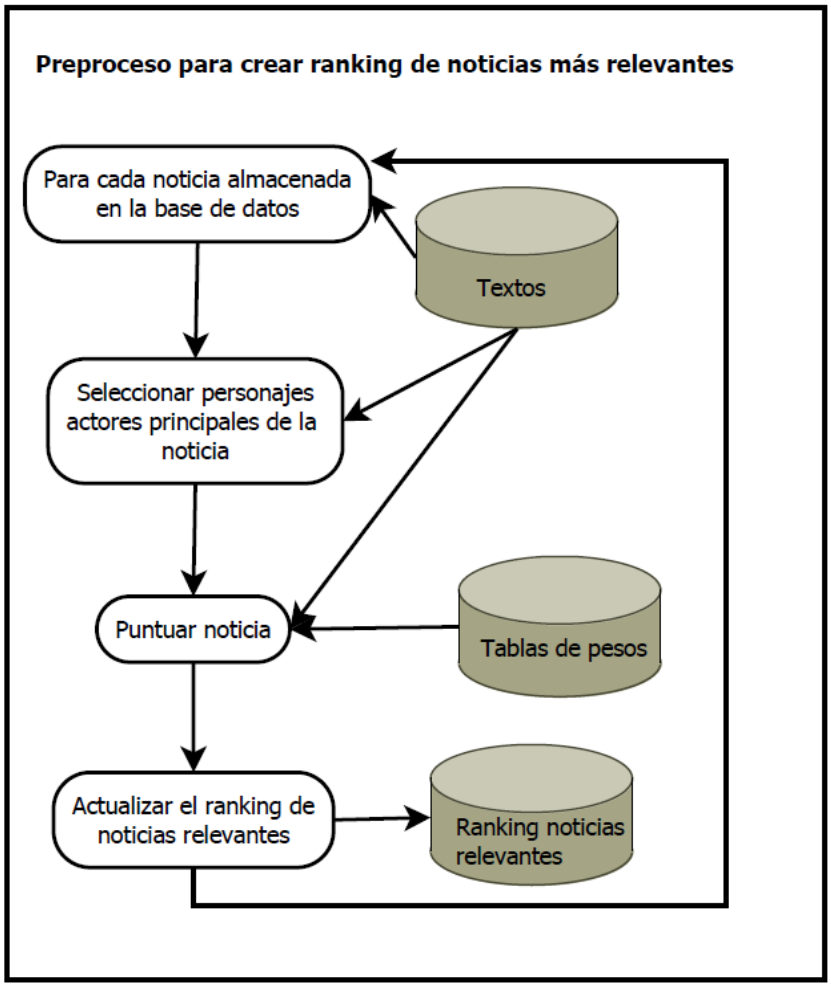

Figura 1.10: Diagrama de actividades para preprocesar noticias más relevantes

En segundo lugar, se han creado unas tablas que asignan determinados pesos para cada intervalo de apariciones del nombre en cada uno de los lugares nombrados antes. Seleccionaremos las noticias de mayor puntuación.

Para ajustar los resultados, se han realizado pruebas con diferentes pesos, y determinar cuáles nos obtienen las noticias deseadas.

La puntuación se almacenará en la base de datos EMMA de tal forma que cada texto almacenado se corresponda con una serie de personajes y su puntuación obtenida en este proceso para cada uno de ellos.

El proceso de valoración de relevancia de noticias se lanzará diariamente por el personal de documentación para mantener la base de datos actualizada.

El desarrollo más detallado de este proceso está descrito en el Anexo B, sección 3.5.

3. ARQUITECTURA DEL SISTEMA

En este capítulo se describe la arquitectura general del generador de fichas biográficas y que se muestra a continuación en la figura 1.11.

El sistema desarrollado se compone de una aplicación diseñada para integrarse dentro de las ya existentes en el escritorio de EMMA, plataforma que gestiona la información en el medio periodístico en el que se desarrolla este PFC y que se explica con más detalle en el Anexo B, apartado “Contexto”.

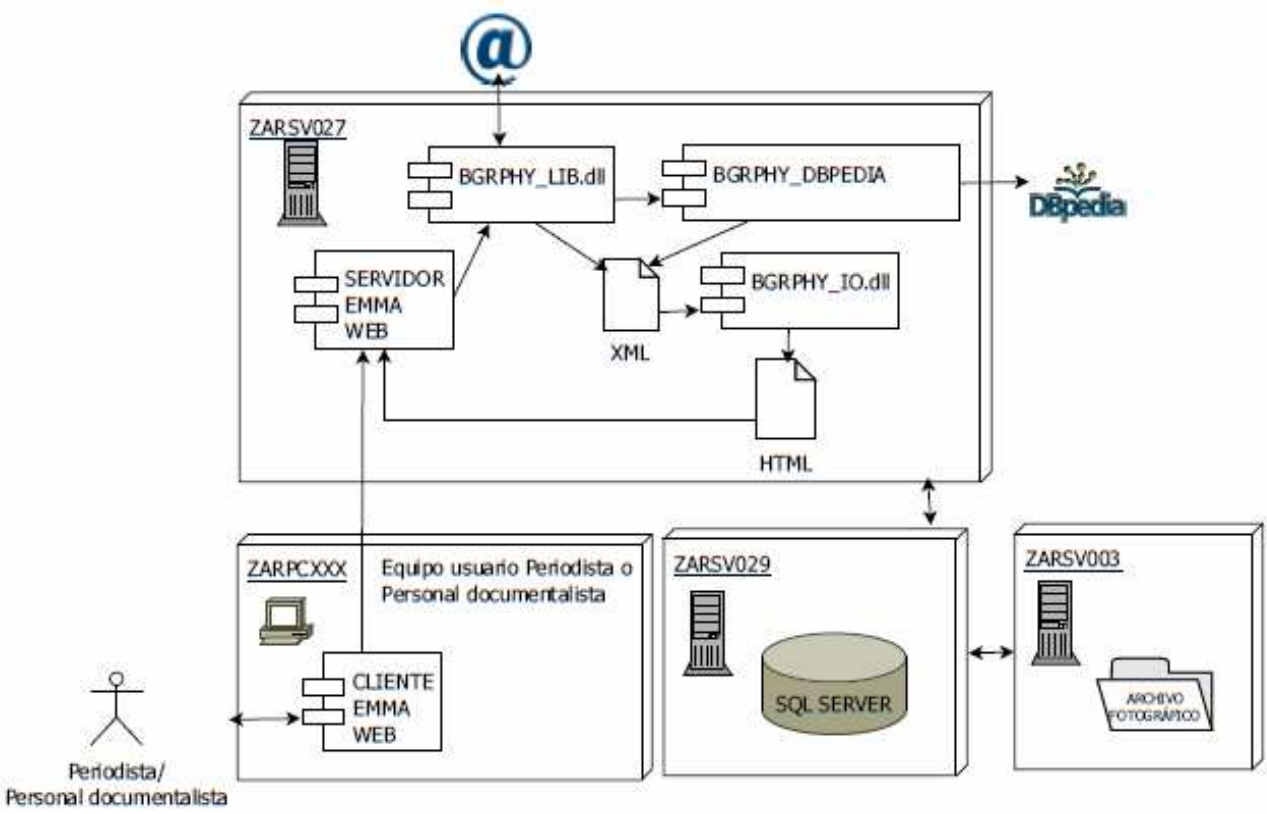

Figura 1.11: Diagrama de despliegue del sistema completo

Para obtener una ficha biográfica en primer lugar, el usuario periodista o documentalista se conectará a través de una interfaz de usuario (BGRPHY_IO), que será la que llame a la segunda librería (BGRPHY_LIB). Esta librería está encargada de realizar las búsquedas en Internet y en el archivo documental (almacenado en una base de datos SQL Server). Se apoyará en otra librería (BGRPHY_DBPEDIA), que realiza las búsquedas en DBpedia. Finalmente, también la primera librería (BGRPHY_IO) será la que dé formato a los resultados para presentarlos al usuario. Estas tres librerías se han desarrollado en este PFC y se comunican entre sí a través de archivos XML.

En el diagrama de actividades del sistema (Fig. 1.12) se muestra el orden en que se van produciendo las distintas operaciones. En él vemos que el sistema realiza en primer lugar una búsqueda previa, para analizar la presencia del personaje en el catálogo *de Named Entities* del archivo documental y en DBpedia. Si no aparece, se descarta la búsqueda del personaje. Si su presencia es relevante (porque aparece en ambos), y hay problemas de ambigüedad (coincidencia de dos o más entidades con el mismo nombre), lo resuelve.

Con un nombre de personaje definido tras esta primera etapa, se comienza la búsqueda de toda la información que formará parte de la ficha biográfica, en primer lugar en el archivo documental (noticias, fotografías), a continuación en Internet (datos biográficos en Wikipedia, enlaces a blogs, webs, libros, otras noticias, vídeos, redes sociales). Finalmente completa otros datos biográficos consultando DBpedia en español.

Con toda esta información se genera un archivo XML que finalmente se transformará en uno HTML para presentación al usuario. Estos dos pasos son necesarios ya que el fichero XML nos da un resultado que es independiente de si se va a mostrar en un solo equipo mediante un visualizador, o si se integra en una plataforma web como es EMMA.

Hay que tener también en cuenta que la aplicación puede ser utilizada por diferentes usuarios a la vez, por lo que hay que realizar cierto control de la concurrencia, que se describe con más detalle en el Anexo B, capítulo 3.

Para poder realizar la búsqueda previa del personaje, para extraer fotografías de primer plano, y para filtrar las noticias más relevantes almacenadas sobre cada personaje, se han tenido que realizar unos trabajos previos imprescindibles para lograr resultados en tiempo real, a los que ya se han aludido en el capítulo anterior y se detallan en el Anexo B, secciones 3.3, 3.4 y 3.5.

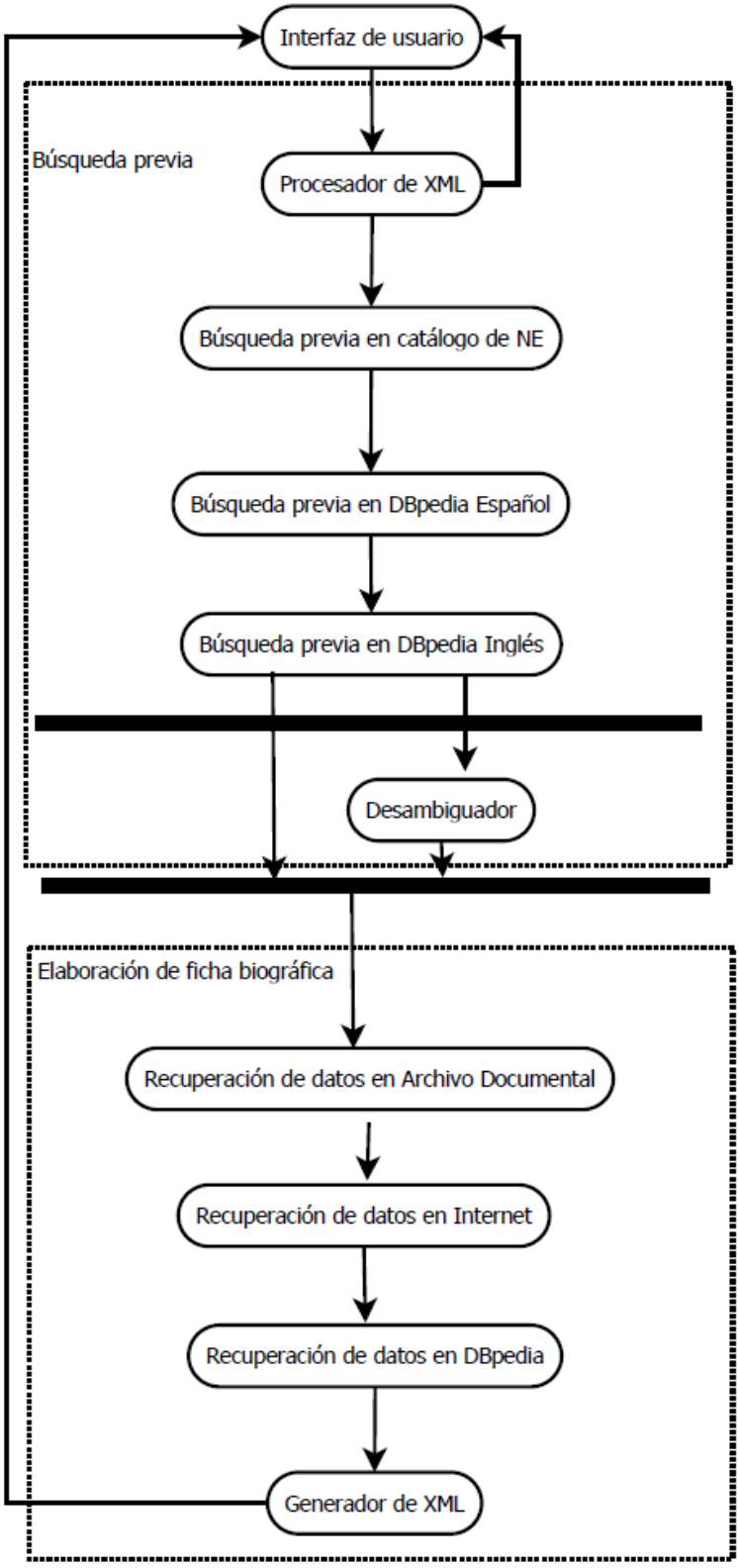

Fig. 1.12 Diagrama de actividades

3.1 La librería BGRPHY_IO. Interfaz del sistema

La librería BGRPHY_IO tiene por un lado, la función de recoger los datos del usuario y trasladarlos a un archivo XML. Por otro, la de mostrar los resultados recogidos en otro archivo XML. Esta librería es la única que puede variar en función del uso definitivo del generador de fichas biográficas. En este caso, se describe tal como está terminado para un uso fuera del entorno de EMMA, es decir, como una herramienta independiente.

Entrada de datos

La recogida de datos varía si el usuario es un periodista o documentalista (estos últimos pueden necesitar delimitar más la búsqueda o los resultados). Por tanto la interfaz inicial debe dar la posibilidad de una búsqueda rápida o avanzada. Otra función de esta librería es la de dar acceso al personal informático a la configuración de la aplicación. Esta interfaz mostrará los datos de configuración y permitirá modificarlos en el correspondiente archivo XML de configuración (Fig. 1.13).

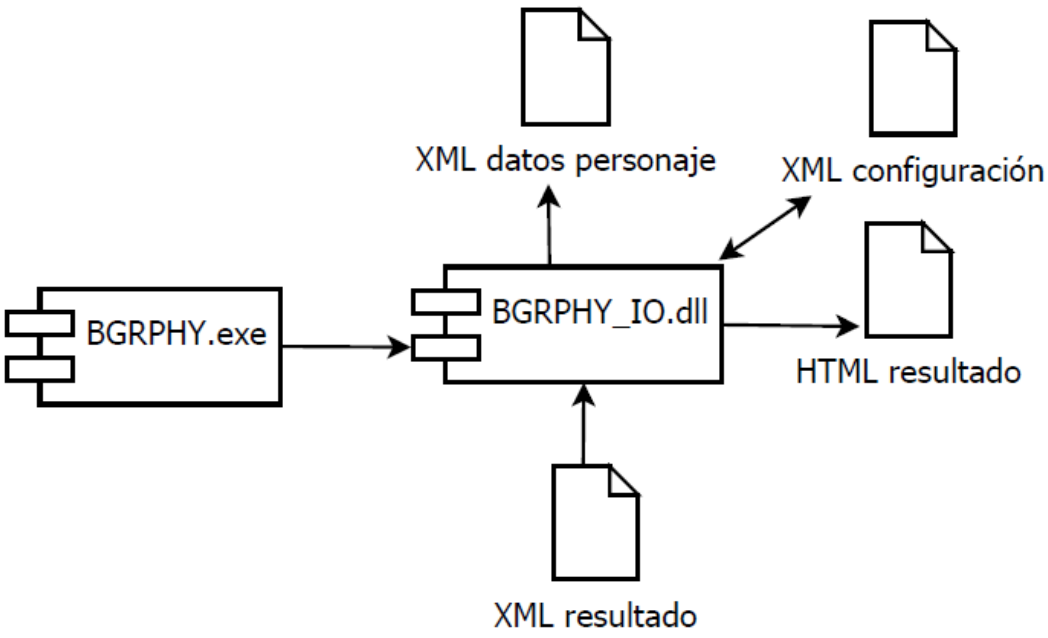

Figura 1.13: Esquema general relativo a la interfaz de entrada-salida.

La interfaz de entrada no varía de la del prototipo, excepto que se añade la funcionalidad de configurar la aplicación (Figs. 1.14, 1.15 y 1.16).

En la pestaña de búsqueda avanzada que vemos en la figura 1.15 es donde el usuario documentalista delimita el número de resultados que aparecen el la ficha biográfica resultante.

Figura 1.14: Pantalla de inicio de la aplicación y recogida de datos. Búsqueda rápida

Figura 1.15: Pantalla inicial para búsqueda avanzada

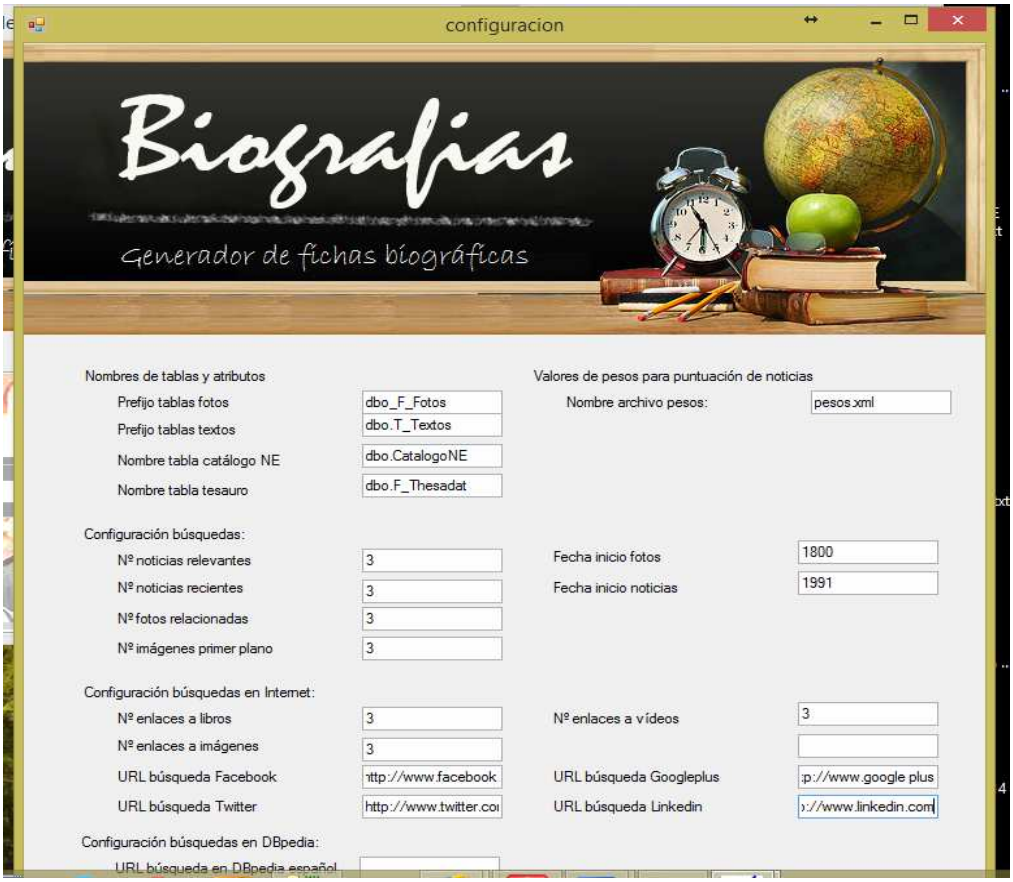

Fig. 1.16: Pantalla de configuración de la herramienta

Salida de datos

En función de los parámetros de configuración, tras la búsqueda se obtiene un archivo XML más o menos extenso, que esta librería transforma en un archivo HTML que mostrará en el navegador. La configuración permite delimitar el número de resultados obtenidos para cada uno de los apartados que forman la ficha biográfica.

A continuación de muestran algunos pantallazos de la ficha resultante (es bastante extensa para mostrarse de una sola vez):

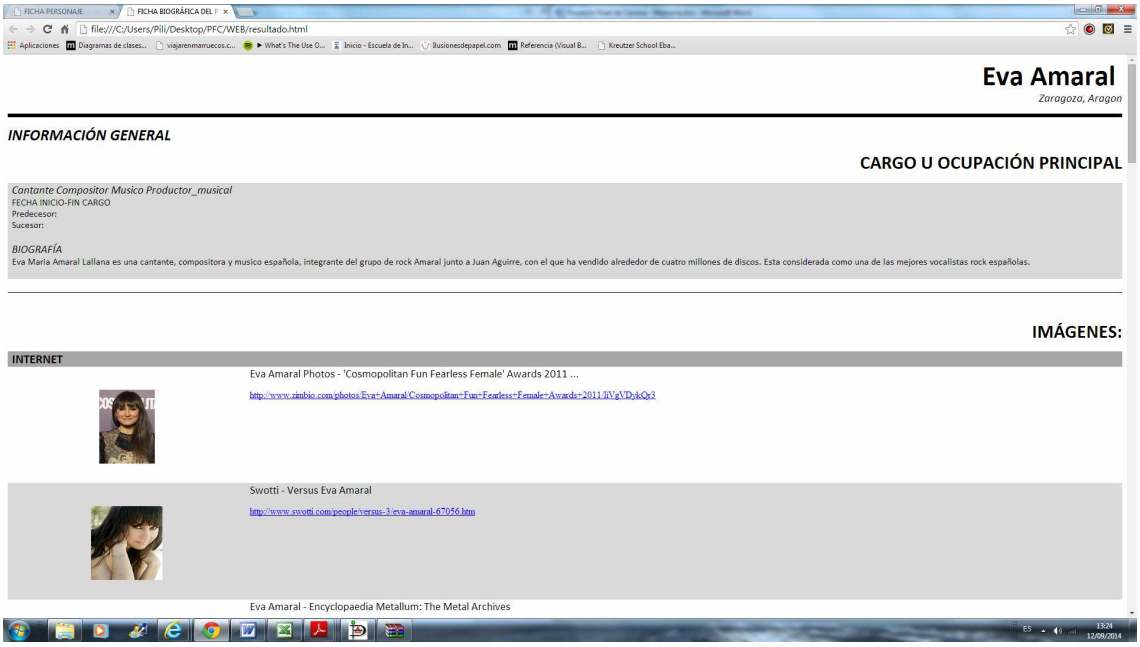

Figura 1.17: Parte superior del archivo HTML resultante

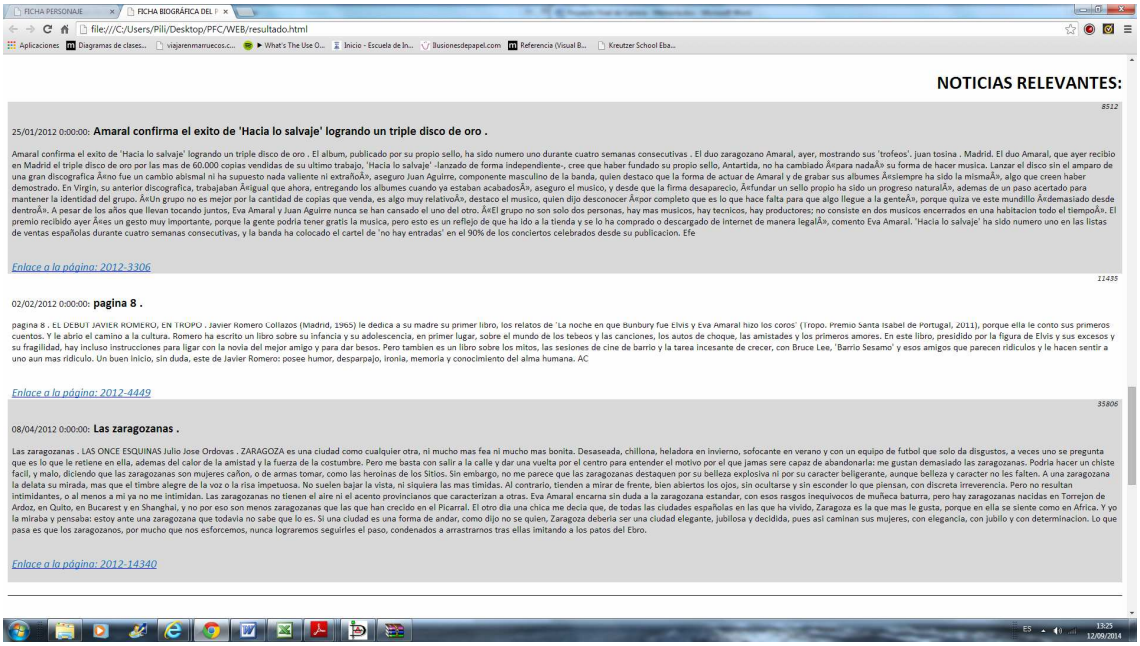

Figura 1.18: Parte central del archivo HTML resultante

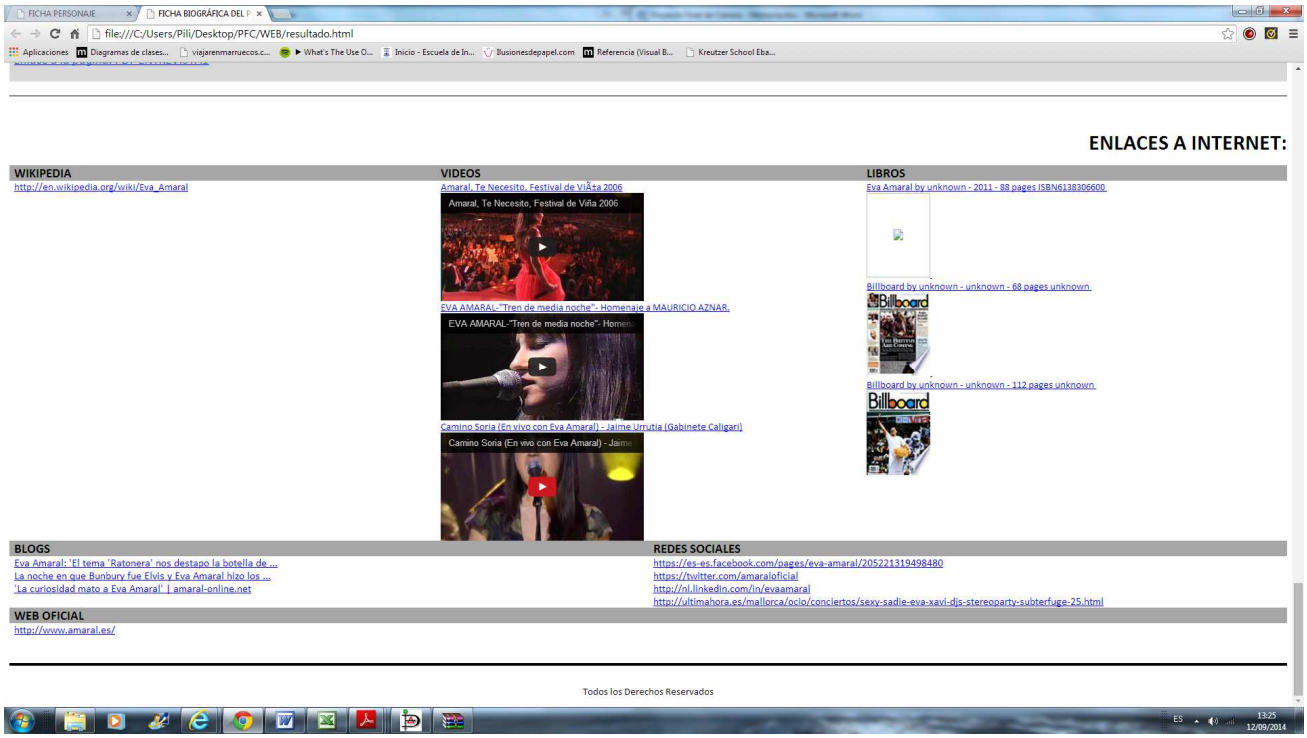

Figura 1.19: Parte inferior del archivo HTML resultante

3.2 La librería BGRPHY_LIB. Búsqueda en el archivo documental e Internet

Esta librería es la que realiza la mayor parte de las funciones que completarán la ficha biográfica. Podríamos dividir estas funciones en dos grandes bloques: la búsqueda previa y la elaboración de la ficha biográfica. A su vez, la elaboración de la ficha tendría otros dos bloques, la recuperación de información del archivo documental, y la recuperación de información de Internet.

En la figura 1.20, se describe la relación entre esta librería con los archivos de entrada y salida, y las fuentes en las que busca la información.

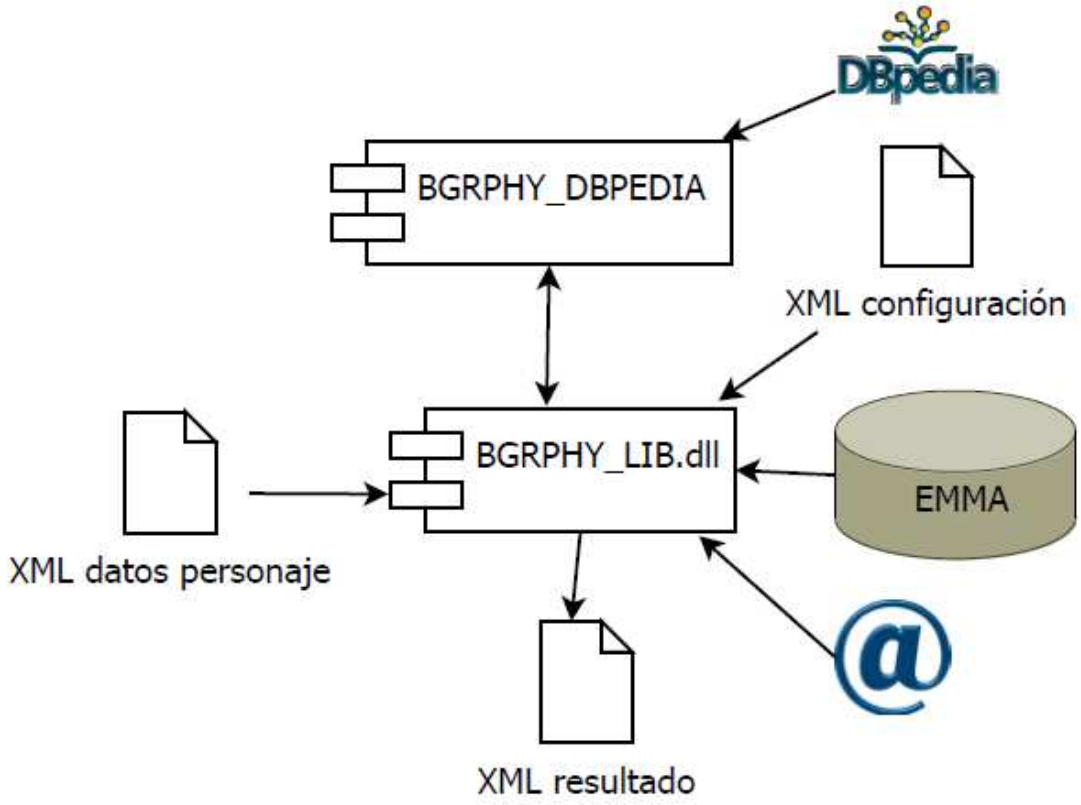

Figura 1.20: Esquema general de componentes para la librería BGRPHY_LIB

Esta librería utiliza la base de datos EMMA para localizar textos e imágenes de archivo sobre el personaje. Valora su actualidad y relevancia, y las incluye en la ficha biográfica. También llama a la librería BGRPHY_DBPEDIA para extraer datos biográficos de DBpedia y añadirlos a la ficha, además de buscar en Internet también vídeos, blogs, redes sociales, libros y otras imágenes que completen la información sobre el personaje buscado.

El archivo XML de configuración nos indicará la extensión del la ficha resultante, y nos dará otros parámetros necesarios para las diferentes búsquedas.

El archivo XML con los datos del personaje nos viene del formulario de recogida de datos que ha rellenado el usuario.

Finalmente esta librería devuelve también un archivo XML que contiene toda la información que formará la ficha biográfica.

A continuación se describe más en detalle el trabajo de la librería BGRPHY_LIB.

1ª Fase: Búsqueda previa

En este bloque se recogen los datos introducidos por el usuario (nombre del personaje y parámetros de configuración) desde un archivo XML inicial. Con estos datos se realiza una encuesta de presencia del personaje en los repositorios donde buscaremos después. Esta función se lleva a cabo para descartar búsquedas de

personajes de los que no tenemos apariciones ni en la base de datos de archivo del periódico, ni en DBpedia (Fig. 1.21).

Con esta búsqueda además, recuperamos el nombre "clave" almacenado en el catálogo de *Named Entities* y que utilizaremos para localizar la información en la base de datos documental.

Si el nombre del personaje recogido del usuario nos puede llevar a más de una entidad, se lleva a cabo un proceso de desambiguación.

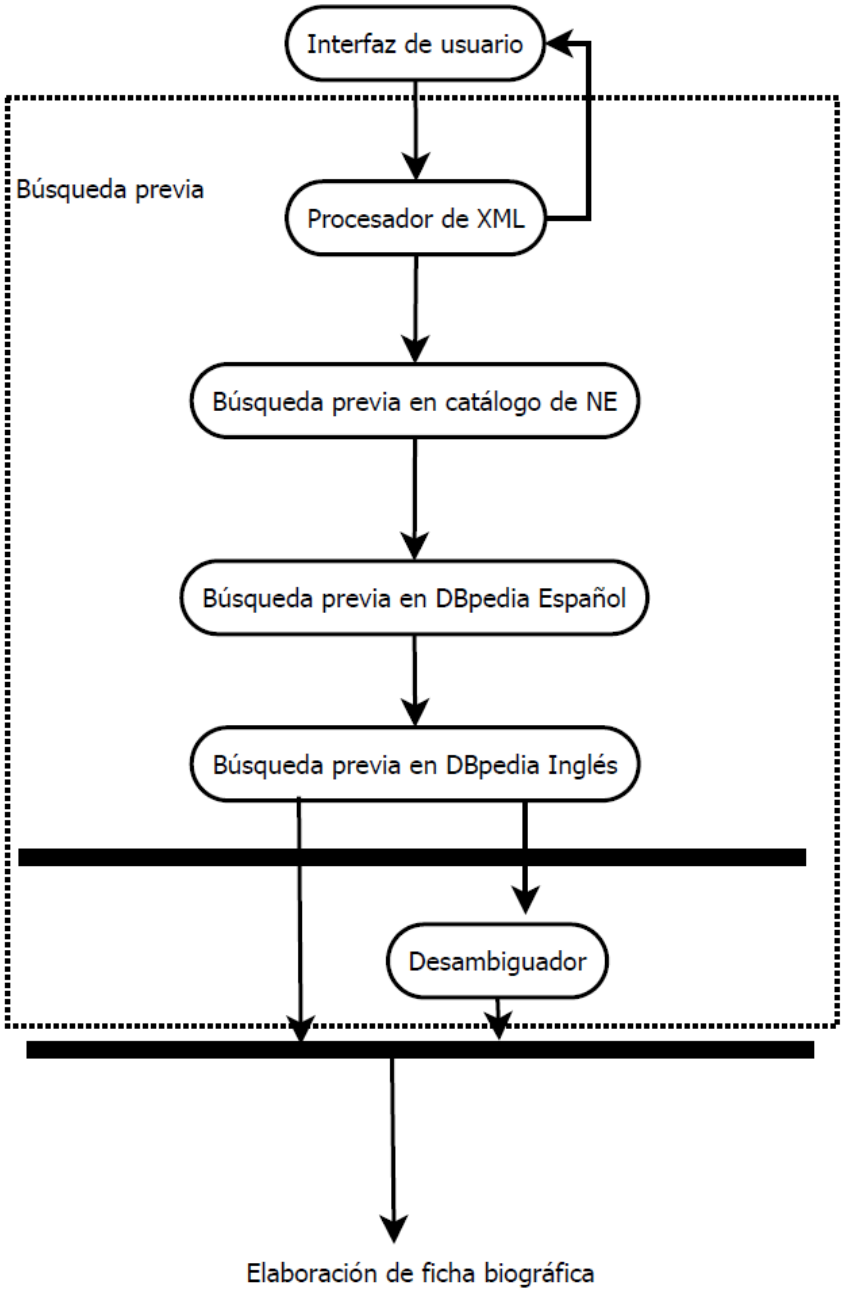

Figura 1.21: Búsqueda previa

El proceso de desambiguación consiste en buscar esa entidad en el catálogo de *Named Entities* creado previamente, y buscarla también en DBpedia en español e inglés. La entidad que aparezca en el catálogo se prioriza. Si no aparecen o aparecen ambas, se busca en DBpedia para discriminar si es tipo *person.* Priorizamos la que lo sea. Si ambas lo son, priorizamos la que aparezca en DBpedia español.

En el caso de que ambas entidades aparezcan en las tres fuentes, y sean personas, tendremos que valorar como más relevante una de ellas buscando el mayor número de apariciones en el campo tesauro y *keywords* de la base de datos EMMA.

2ª Fase: Elaboración de la ficha biográfica

En este bloque de funciones, con el personaje a buscar ya definido, se pasa a recopilar la información deseada sobre él. Las fuentes son la base de datos del periódico (de donde se extraen fotografías recientes, fotografías de primer plano, noticias recientes, noticias más relevantes y entrevistas), Internet (de donde se extraen enlaces a imágenes, vídeos, blogs, libros, Wikipedia, web oficial, Facebook, Twitter, Linkedin, Google+) y DBpedia (de donde se extrae una biografía, cargos o profesión, fecha y lugar de nacimiento y muerte, predecesor y sucesor en el cargo).

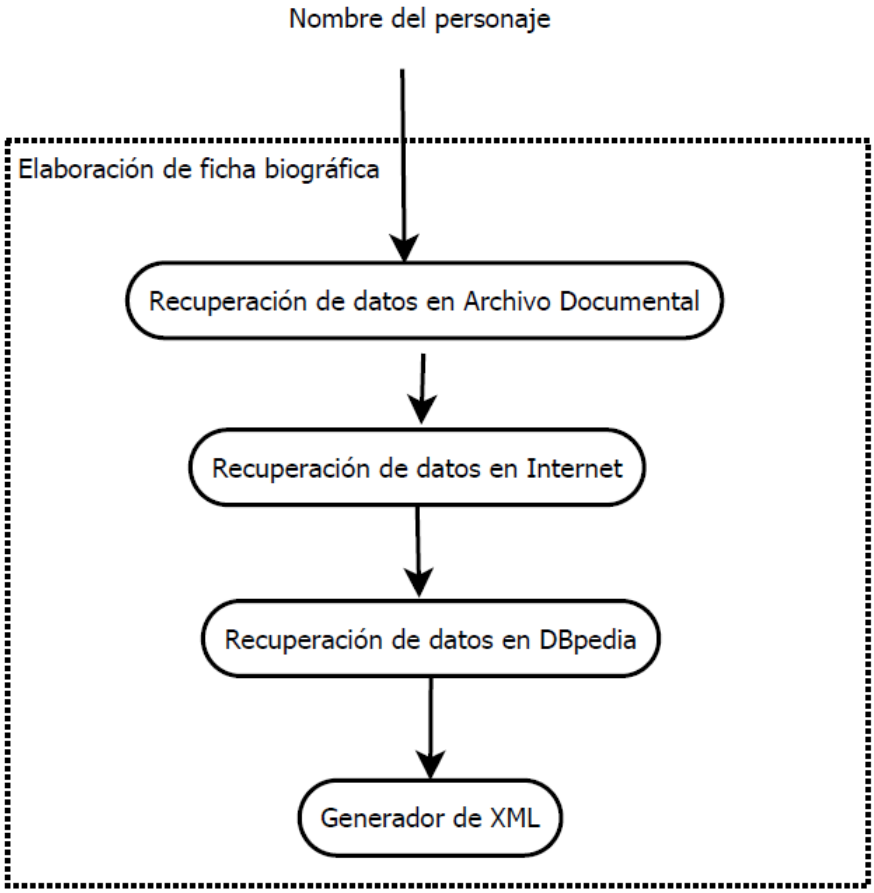

Figura 1.22: Elaboración de la ficha biográfica

Recuperación de datos en el archivo documental

En esta primera etapa de la elaboración de la ficha biográfica, se accede a la base de datos EMMA para recuperar la información relevante. Se extraen los textos de las noticias más recientes y más relevantes (ver sección 2.4.3 de esta misma memoria) que ya están previamente etiquetadas para los personajes que tenemos en el catálogo de *Named Entities*. Estos textos se completan con las imágenes en PDF de las páginas originales del periódico donde se publicaron, y con fotografías que están almacenadas en los servidores ZARSV002/003, en las rutas que recuperamos de EMMA (ver Fig. 1.23).

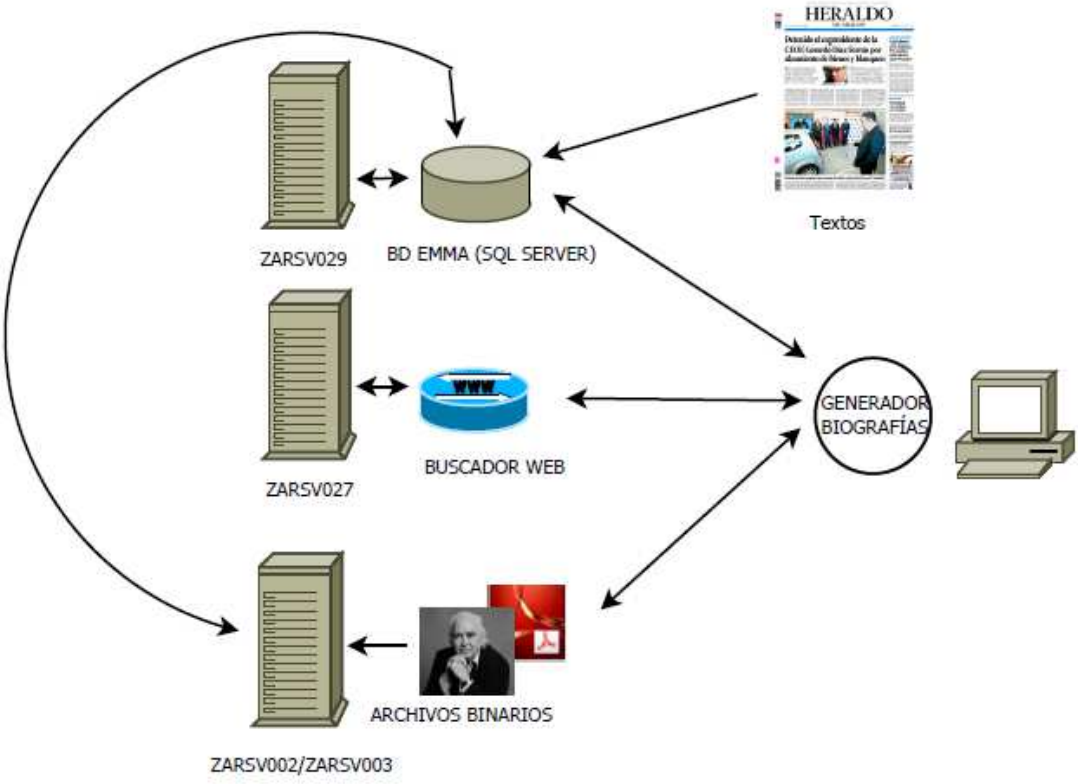

Figura 1.23: Sistema de almacenamiento de la información en el medio periodístico.

Recuperación de datos en Internet

Para la recuperación de datos en Internet, se han utilizado APIs de Google, opción que se tomó después de quedar obsoleto el web service de Bing que utilizaba en el prototipo. Son unas APIs de uso muy extendido, que cuentan con librerías para .NET que es el lenguaje que se utiliza en esta herramienta.

Para la búsqueda de vídeos se ha utilizado la última librería proporcionada por Google, que es YouTube API V3. Ésta nos permite, con sólo tenernos identificados y de manera gratuíta, una cuota de 10.000 de operaciones de lectura por día. Por el uso que se va a dar a esta herramienta, es más que suficiente.

Para el resto de búsquedas, se ha utilizado la librería Google API Search que nos permite obtener resultados de Google Search, Google Books, Google Blogger, Google

Images, y Google News. Las búsquedas nos permiten también obtener los resultados disponibles de los perfiles de los personajes en las redes sociales. Esta librería utiliza otras a su vez oficiales de Google y funciona del mismo modo que la librería para YouTube, es decir, con el mismo tipo de identificación (a través del correo de Gmail), y es gratuita mientras no se exceda el número de consultas diarias permitidas (10,000).

Estas librerías nos devuelven unos objetos que representan los resultados en Google: libros, vídeos, imágenes, etc. que tienen unos atributos característicos que nos permiten recuperar su URL, su título, *thumbnails*, etc.

3.3 La librería BGRPHY_DBPEDIA. Búsqueda en DBpedia

Finalmente paso a describir la última fuente de información utilizada en este PFC y relativa a la Web Semántica, que es DBpedia. Los estudios previos relativos a este tema que han sido necesarios para poder incluir esta fuente de información en el proyecto, se recogen en el Anexo A, sección A.3.

Esta librería es la que da acceso a la información contenida en DBpedia y utiliza otra librería ampliamente difundida, *dotnetrdf.dll*. Esta librería ofrece una serie de funciones a las que, proporcionándoles el texto de la consulta SPARQL y la URL del *endpoint*[7] donde se realiza la consulta, permite conectarse con fuentes que tengan una interfaz SPARQL y obtener los resultados pertinentes.

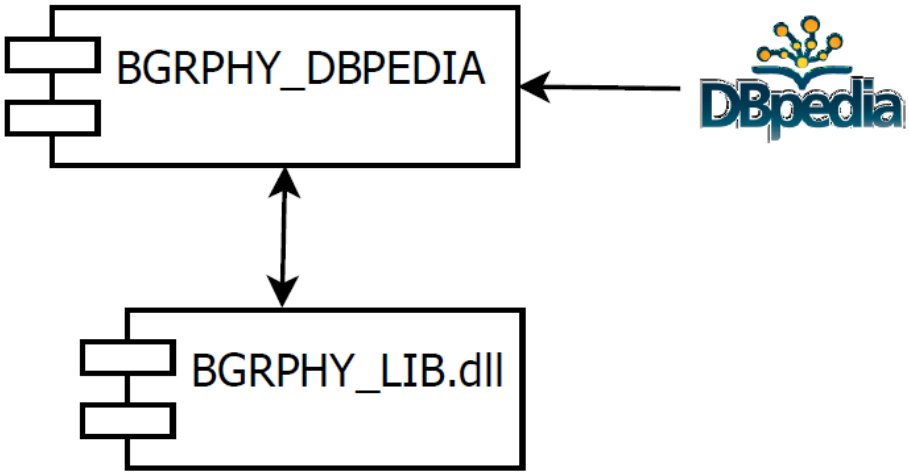

Figura 1.23: Módulo de acceso a DBpedia

La librería BGRPHY_DBPEDIA utiliza el nombre del personaje que le proporciona BGRPHY_LIB y construye la consulta SPARQL que en nuestro caso, busca recoger los atributos del recurso que lleva ese nombre. Utiliza la librería *dotnetrdf.dll* para lanzar la consulta. Si el recurso existe, recorre sus atributos y recoge y devuelve los que coinciden con los tipos que hemos definido previamente: fecha de nacimiento y muerte, lugares de nacimiento y muerte, cargos, predecesor, sucesor y un resumen de la biografía.

[7] Un *endpoint* permite a un usuario realizar una consulta, en este caso usando el lenguaje SPARQL a un repositorio de información.

Para algunos de estos datos podemos encontrar diferentes nombres de atributo que se han contemplado, dependiendo de si buscamos en la DBpedia inglesa o española. Incluso existen diferentes nombres de atributo en una misma DBpedia que se consultan en este módulo para intentar recoger siempre información (en algunas entidades aparece por ejemplo el atributo lugar de nacimiento como *dbpedia-owl:birthPlace,* en otras como *prop-es:lugarNacimiento*). Como inconveniente se ve que ya que la DBpedia española es un proyecto aún en desarrollo, es común no encontrar almacenadas sobre todo, las fechas.

Los datos que se obtienen de DBpedia son los que se extraerían mediante técnicas de PLN como los descritos en la sección 2.6 del Anexo B. En la práctica estas técnicas no se han implementado, pero servirían para encontrar los datos biográficos de personajes que aún no se encuentran representados como un recurso en DBpedia.

4. METODOLOGÍA Y RESULTADOS

4.1 Metodología de desarrollo

El desarrollo se corresponde con el patrón clásico de análisis, diseño, implementación, pruebas y puesta en marcha, siguiendo una metodología de desarrollo en cascada debido a dos factores fundamentales. En primer lugar se trata de un sistema de tamaño medio-pequeño y en segundo sólo trabaja una persona directamente sobre él.

Sin entrar en detalles sobre el análisis y el diseño que se explican en el Anexo B, puede resultar interesante mencionar cómo se realizó la implementación y puesta en marcha.

En primer lugar, hay que destacar la importancia que tuvo en el desarrollo de este proyecto el tiempo y esfuerzo dedicados a los estudios previos. Para poder abordar la realización de esta herramienta, se dedicó una parte importante de la planificación global en analizar libros y artículos de investigación que aportaran técnicas en el uso del "text mining" y la desambiguación de nombres, entre otros temas relacionados con el procesamiento del lenguaje natural. Por otro lado, el reconocimiento facial fue el segundo tema de estudio, para poder localizar fotos de primer plano dentro del archivo documental del periódico (ya que el sistema existente no tiene etiquetadas como tales las fotos de primer plano). Finalmente, hubo que profundizar en el conocimiento de la Web Semántica, *"linked data"* y temas relacionados, que ayudaran a utilizar un repositorio semántico como es DBpedia, y poder consultarlo utilizando lenguajes como SPARQL.

Entrando ya en el desarrollo en sí, tras un primer análisis, se realizó un prototipo que nos permitió analizar el acceso a la base de datos de archivo e Internet. Este prototipo fue usado para una primera batería de pruebas, de la que surgen errores y sugerencias que permitieron afinar el funcionamiento del sistema y, sobre todo, tener en cuenta los tiempos necesarios de búsqueda que nos limitan.

A continuación se entró ya en el análisis y diseño definitivos, que se incluyen en el Anexo B. La implementación definitiva incluye las mejoras en la búsqueda, añadiendo además la librería de búsqueda en DBpedia, la búsqueda de fotografías y otras búsquedas en Internet. Finalmente se elaboró la librería que gestiona la interfaz y que permitió utilizar la herramienta a los usuarios para hacer las pruebas.

En el Anexo B, sección 2.4 (desarrollo y pruebas) se pueden encontrar las pruebas realizadas con mayor detalle.

Tras la realización de una segunda batería de pruebas por los usuarios reales, en este caso personal de documentación y periodistas, se mantuvo un diálogo abierto para corregir problemas y errores, y finalmente, se puso la versión más estable a disposición de los usuarios.

4.2 Resultados

Para la evaluación general de los resultados de la herramienta, se ha realizado una pequeña encuesta en un grupo formado por 100 trabajadores del Grupo Heraldo. De ellos 20 pertenecen al Departamento de Documentación y por otro lado, 80 son periodistas o usuarios con experiencia en tareas de documentación.

En esta encuesta se presentó el generador de fichas biográficas y se cuestionó sobre la utilidad que encuentran en la generación de fichas sobre personajes de forma automática y la posibilidad de obtener simultáneamente datos del almacenamiento interno al periódico y de Internet. Después de probar la herramienta, se pidió una puntuación de 1 a 10 sobre la importancia que le dan para el desarrollo de su trabajo (REL), la utilidad (UTIL), la facilidad de manejo (FAC) y por último la novedad (NOV) entre las herramientas que utilizan habitualmente. El resultado puede verse en la tabla siguiente:

USUARIO	ITEMS A VALORAR				
PERIODISTA	REL	UTIL	FAC	NOV	TOTAL
Elaboración de fichas automáticas	9	9	9	10	9,25
Búsqueda conjunta en Intranet e Internet	8	9	8	9	8,5
DOCUMENTALISTA					
Elaboración de fichas automáticas	9	9	7	10	8,75
Búsqueda conjunta en Intranet e Internet	9	10	9	8	9

En los resultados se puede apreciar que la elaboración de fichas automáticas es mejor valorada por los periodistas, que por el personal documentalista que prefiere controlar más el contenido de las búsquedas. Sin embargo, estos últimos valoran más la facilidad de recoger información simultáneamente de la base de datos EMMA e Internet.

5. CONCLUSIONES

En este capítulo se describen los resultados que hemos obtenido después del desarrollo y pruebas del generador de fichas biográficas incluyendo futuras ampliaciones y una valoración personal de todo el proceso.

5.1 Resultados obtenidos

Repasando los objetivos marcados al principio del documento, se puede concluir que, respecto a la investigación en técnicas de procesamiento del lenguaje natural, se ha hecho un esfuerzo en ofrecer una visión del estado del arte y se han encontrado soluciones que se han aplicado en el desarrollo del proyecto.

Respecto a la desambiguación, es un problema que se resuelve en la implementación de la herramienta, aunque es mejorable, como se indica en la sección de futuras ampliaciones que aparece a continuación.

En relación a las fotografías de primer plano, se ha añadido una funcionalidad que no existía previamente en el entorno de trabajo del medio periodístico, ya que un módulo que identifica este tipo de fotos se ha aplicado en el etiquetado de fotografías de todo el archivo documental y no sólo de las utilizadas en las fichas biográficas.

Se ha hecho un análisis del estado de la Web Semántica, utilizando un repositorio semántico como DBpedia para solucionar diferentes requisitos del sistema.

Finalmente se ha implantado una herramienta que proporciona tanto a documentalistas como periodistas, una ficha biográfica sobre un personaje que les ofrece una información completa sobre él, reduce sensiblemente el tiempo de localizar la información y facilita su trabajo, con lo que se considera cumplido el objetivo inicial.

Añadir además que todo el trabajo de este proyecto ha dado lugar a un artículo de investigación desarrollado con el grupo SID, en cuya elaboración he participado colaborando como un autor más. Será presentado a la conferencia internacional CAISE 2015 (CORE A) que se celebrará en Estocolmo en Junio de 2015. En el anexo C se adjunta el artículo.

5.2 Valoración personal

Supongo que esta no es una valoración al uso, ya que en mi caso, han pasado dieciséis años desde que terminé la última asignatura hasta que por fin estoy terminando estas líneas. El aprendizaje durante la realización del proyecto ha sido enorme, ya que he podido comprobar la evolución tan brutal que sufre el campo de la informática. Y además con la suerte de poder entrar a conocer campos de futuro como son la Web Semántica, las futuras búsquedas más efectivas en Internet, el campo de las ontologías y la organización de la información.

Para mi ha supuesto en primer lugar, recuperar conocimientos ya adquiridos, y a continuación, aprender estándares nuevos como XML, SPARQL, UML…, adquirir más experiencia en .NET y Windows (hemos utilizado servicios Windows, servicios Web, aplicaciones, servidores de bases de datos), y descubrir en la práctica lo que es la Ingeniería del Software.

El tiempo dedicado al estudio y aprendizaje de técnicas y algoritmos ha sido muy enriquecedor, y la búsqueda de artículos de investigación y su aplicación ha sido, para mí, lo más interesante académicamente.

Sin embargo, humanamente, el poder trabajar en un equipo como el de Ibercentro Media y el grupo SID, y en un ambiente como es el de la plantilla del Heraldo de Aragón, es una experiencia de aprendizaje única, ya que he podido vivir en directo cómo afecta mi forma de analizar y realizar un proyecto en un equipo de personas que lo están demandando y utilizando.

El participar de la realización del artículo de investigación (ver Anexo C) para una conferencia internacional, me ha enriquecido mucho, acercándome a un terreno que para mi era totalmente desconocido como es el de la investigación. El formar parte de un grupo como es SID, y poder conocer (si se publica este artículo) cómo es la participación en una conferencia de este tipo me parece muy interesante ya que se asiste a un importante foro de intercambio de ideas.

5.3 Futuras ampliaciones

El generador de fichas biográficas siempre puede ampliarse añadiendo más datos sobre el personaje (como obras realizadas, cronología de noticias, otros datos biográficos) a extraer de las fuentes de información.

Para mejorar el tiempo de búsqueda se pueden encontrar otros algoritmos para analizar la información contenida en el archivo documental. En este mismo sentido, la desambiguación, o las técnicas de *text mining* utilizadas pueden ser otras que se descubran más eficientes. La desambiguación cuando se está buscando en el archivo documental podría hacerse también incluyendo opciones en las que pudiera elegir el propio usuario.

El sustituir decisiones “automáticas” que toma el sistema por valoraciones del usuario, haría que éste pudiera “aprender” y mejoraría en aspectos como el mencionado anteriormente de la desambiguación y otros como la valoración de la relevancia de las noticias.

Respecto al catálogo creado de *Named Entities*, éste puede ser ampliado con más entidades, y utilizar técnicas de desambiguación para evitar redundancias en la extracción de nombres del archivo documental. También el filtrado de nombres puede mejorarse con nuevos algoritmos que eviten “ruido” de NE que no corresponden a personajes como calles, nombres de museos, etc.

En este proyecto, las búsquedas en diferentes fuentes (archivo documental, Internet, DBpedia) se realizan de forma secuencial. Una mejora sustancial sería una búsqueda en paralelo que redujera drásticamente el tiempo empleado. También se pueden buscar datos en otras fuentes: usar otro buscador que mejore los resultados de

Google, o usar otras fuentes de *Linked Data* como Freebase[8] u OpenCyc[9].

Y finalmente, la búsqueda en determinados lugares de Internet para la obtención de imágenes, blogs, libros, u otros datos puede ser susceptible de ser cambiado por otros que sean más actuales o mejoren las búsquedas. Incluir en la búsqueda de redes sociales un "análisis del sentimiento" o valoración del personaje en ellas, añadiría un dato importante a la ficha biográfica.

[8] https://www.freebase.com/
[9] http://www.cyc.com/platform/opencyc

BIBLIOGRAFÍA

Herramientas
[1] Milenium:
http://www.protecmedia.com/es/solutions/milenium-cross-media/visiongeneral
[2] Windows: http://es.wikipedia.org/wiki/Microsoft_Windows
[3] Framework .Net: http://es.wikipedia.org/wiki/Microsoft_.NET
http://www.microsoft.com/net/
[4] Visual Basic .Net: http://es.wikipedia.org/wiki/Visual_Basic_.NET
http://msdn.microsoft.com/es-es/library/2x7h1hfk(v=vs.80).aspx
[5] Visual Studio: http://es.wikipedia.org/wiki/Microsoft_Visual_Studio
[6] Sql Server: http://es.wikipedia.org/wiki/Microsoft_SQL_Server
[7] Java. http://es.wikipedia.org/wiki/Java_(lenguaje_de_programaci%C3%B3n)
http://www.java.com/es/about/
[8] Eclipse: http://www.eclipse.org/
[9] Ubuntu: http://www.ubuntu.com/
[10] Freeling: http://nlp.lsi.upc.edu/freeling/
[11] Gazetteer: Diccionario geográfico. Ver [GABUILME13].

Estándares
[11] SQL: http://es.wikipedia.org/wiki/SQL
[12] XML: http://www.w3.org/XML/
[13] RDF: http://www.w3.org/RDF/
[14] SPARQL: http://www.w3.org/TR/rdf-sparql-query/
[18] HTML: http://www.w3.org/html/

Otros
[15] Web Service: http://www.w3.org/standards/webofservices/
[16] Servicio Windows: http://msdn.microsoft.com/en-us/library/d56de412.aspx
[17] DLL: http://msdn.microsoft.com/en-us/library/windows/desktop/ms682589.aspx
[18] Linked data: http://linkeddata.org/

Referencias bibliográficas:

Procesamiento del lenguaje natural

[CAR97] Empirical Methods in Information Extraction, C. Cardie, 1997. Department of Computer Science Cornell University, IIthaca, NY 14850

[GRISH97] Information Extraction: Techniques and Challenges. Ralph Grishman, 1997. Computer Science Department. New York University, New York, NY 10003, U.S.A.

[NAGHAS98] Automatic Text Summarization Based on the Global Document Annotation. Katashi Nagao, Sony Computer Science Laboratory Inc., KSiti Hasida, Electrotechnical Laboratory, 1998

[SABU88] Salton G, Buckley C (1988). "Term-weighting approaches in automatic text

retrieval".

[SAMCG86] Salton G; McGill MJ (1986). Introduction to modern information retrieval. McGraw-Hill. ISBN 0-07-054484-0 (algoritmo TF IDF)

[LUHN58] Luhn, H. P. (1958). Auto-encoding of documents for information retrieval systems. IBM Research Center.

[GABUILME13] Angel Luis Garrido and Maria G. Buey and Sergio Ilarri and Eduardo Mena, "GEO-NASS: A Semantic Tagging Experience from Geographical Data on the Media", 17th East-European Conference on Advances in Databases and Information Systems (ADBIS 2013), Genoa (Italy), Springer Verlag LNCS, ISSN 0302-9743, ISBN 978-3-642-40682-9, volume 8133, pp. 56-69, September 2013.

[BUPA06] Bunescu, R., & Pasca, M. (2006, April). Using encyclopedic knowledge for named entity disambiguation.

[BIRKLE09] BIRD, Steven; KLEIN, Ewan; LOPER, Edward. Natural language processing with Python. " O'Reilly Media, Inc.", 2009.

[COD95] CODINA, L. L. Teoría de recuperación de información: modelos fundamentales y aplicaciones a la gestión documental. Information World en español, 1995.

[SOD99] SODERLAND, Stephen. Learning information extraction rules for semi-structured and free text. Machine learning, 1999, vol. 34, no 1-3, p. 233-272.

[BLAIR90] BLAIR, David C. Language and representation in information retrieval. 1990.

[MAN99] MANNING, Christopher D. Foundations of statistical natural language processing. MIT press, 1999.

[LLOPA11] COMPENDIUM: Una herramienta de generación de resúmenes Modular.Elena Lloret y Manuel Palomar. Dept. Lenguajes y Sistemas Informáticos, Universidad de Alicante. 2011

[BOSAGG11] Spanish Text Simplification: An Exploratory Study
Stefan Bott, Horacio Saggion. Universitat Pompeu Fabra, Barcelona. 2011

[PLADI11] Using Semantic Graphs and Word Sense Disambiguation Techniques to Improve Text Summarization
Laura Plaza y Alberto Díaz. Universidad Complutense de Madrid. 2011

[CAMARPA03] A Simple Named Entity Extractor using AdaBoost
Xavier Carreras, Lluís Márquez, and Lluís Padró. TALP Research Center. Departament de Llenguatges i Sistemes Informàtics, Universitat Politècnica de Catalunya, 2003

[NAV09] NAVIGLI, Roberto. Word sense disambiguation: A survey. ACM Computing Surveys (CSUR), 2009, vol. 41, no 2.

[DEHO02] Deepak Ravichandran and Eduard Hovy. 2002. Learning surface text patterns for a Question Answering system. In Proceedings of the 40th Annual Meeting

on Association for Computational Linguistics (ACL '02). Association for Computational Linguistics, Stroudsburg, PA, USA, 41-47.

[COETWE01] Cody Kwok, Oren Etzioni and Daniel S. Weld. 2001. Scaling question answering to the web. ACM Trans. Inf. Syst. 19, 3 (July 2001), 242-262.

Reconocimiento facial

[FERET99] The FERET Evaluation Methodology for Face-Recognition Algorithms. P. Jonathon Phillips, Hyeonjoon Moon, Syed A. Rizvi, Patrick J. Rauss, 1999

[BRUPO93] BRUNELLI, Roberto; POGGIO, Tomaso. Face recognition: Features versus templates. IEEE transactions on pattern analysis and machine intelligence, 1993, vol. 15, no 10, p. 1042-1052.

[CAS03] CASTELLS, Pablo. La Web Semántica. Sistemas interactivos y colaborativos en la web, 2003, p. 195-212.

Web Semántica

[SHI03] Shiguihara Juárez, Pedro. "Un breve panorama sobre las Bases de Datos Semánticas". Universidad Nacional de Trujillo, Peru, 2003

[HARLANG10] A Database Perspective on Consuming Linked Data on the Web. Olaf Hartig, Andreas Langegger, 2010

[GARMAR07] Ontologías y organización del conocimiento: retos y oportunidades para el profesional de la información, 2007 Francisco-Javier García-Marco

[GRU93] Thomas R. Gruber. 1993. A translation approach to portable ontology specifications. Knowl. Acquis. 5, 2 (June 1993), 199-220.

[HAR11] SPARQL for a Web of Linked Data: Semantics and Computability, 2011. Olaf Hartig

[HARBIZFREY09] Executing SPARQL Queries over the Web of Linked Data, 2009. Olaf Hartig, Christian Bizer, and Johann-Christoph Freytag

[RISYUATH00] RISHE, Naphtali, et al. Semantic access: semantic interface for querying databases. En VLDB. 2000. p. 591-594.

[GARGA11] Garcia, M., & Gamallo, P. (2011). Resolución de Correferencia de Nombres de Persona para Extracción de Información Biográfica. Procesamiento de Lenguaje Natural, 47, 47-55.

[CPC12] del Cisne Garcıa, M., Pasmay, F., & Carrera (2012), Un algoritmo simple y eficiente para la clasificación automática de páginas web.

Tesauros

[CACU07] Castillo, Lourdes; Cueva, Alejandro de la. “Evolución y uso de los lenguajes controlados en documentación informativa”.
En: El profesional de la información, 2007, noviembre-diciembre, v. 16, n. 6, pp. 617-626.

ANEXO A: ESTUDIOS PREVIOS

Para la elaboración de este PFC ha sido muy importante una fase previa de estudio de diferentes documentos, que han hecho posible el desarrollo de algunos de los componentes de la herramienta.

Para conocer cómo extraer información de los textos no estructurados almacenados tanto en la base de datos de archivo (EMMA) como en distintos repositorios de Internet, ha sido necesario realizar un estudio del arte previo y utilizar diferentes artículos de investigación relacionados con el procesamiento del lenguaje natural, la extracción de datos y Named Entities, resolución de correferencias, algoritmos de reconocimiento facial y obtener información general respecto la Web Semántica y su uso práctico.

A1. PROCESAMIENTO DEL LENGUAJE NATURAL

A1.1 Búsqueda y recuperación de información

La **Búsqueda y Recuperación de Información**, llamada en inglés *Information Search and Retrieval* (ISR), es la ciencia de la búsqueda de información en documentos electrónicos y cualquier tipo de colección documental digital, encargada de la búsqueda dentro de éstos mismos, búsqueda de metadatos que describan documentos, o también la búsqueda en bases de datos relacionales, ya sea a través de Internet, o intranet, y como objetivo realiza la recuperación en textos, imágenes, sonido o datos de otras características, de manera pertinente y relevante.

La recuperación de información es un estudio interdisciplinario. Cubre tantas disciplinas que eso genera normalmente un conocimiento parcial desde tan solo una u otra perspectiva. Algunas de las disciplinas que se ocupan de estos estudios son la psicología cognitiva, la arquitectura de la información, diseño de la información, inteligencia artificial, lingüística, semiótica, informática, biblioteconomía, archivística y documentación.

Para alcanzar su objetivo de recuperación se sustenta en los sistemas de información, y al ser de carácter multidisciplinario intervienen bibliotecólogos para determinar criterio de búsqueda, la relevancia y pertinencia de los términos, en conjunto con la informática.

La idea del uso de computadoras para la búsqueda de trozos relevantes de información se popularizó a raíz de un artículo *As We May Think* de Vannevar Bush en el año 1945. Los primeros sistemas automatizados de recuperación de la información fueron presentados durante la década de 1950 a 1960. Durante 1970 se realizaron pruebas en un grupo de textos como la colección Cranfield para un gran número de distintas técnicas cuyo rendimiento fue bueno. Los sistemas de recuperación a larga escala, como el Sistema de Diálogo Lockheed, comenzaron a utilizarse a principios de

1970.

En 1992, el Departamento de Defensa de los Estados Unidos conjuntamente con el Instituto Nacional de Standards y Tecnología (NIST), patrocinaron la Conferencia de Recuperación de Texto (TREC) como parte del programa TIPSTER[10]. Esto proveyó ayuda desde la comunidad de recuperación de la información al suministrar la infraestructura necesaria para la evaluación de metodologías de recuperación de texto en una colección a larga escala. La introducción de motores de búsqueda ha elevado aún más la necesidad de sistemas de recuperación con mayor capacidad.

Algunos de los estudiosos más destacados dentro de esta subdisciplina son Gerard Salton, W Bruce Croft, Karen Spärck Jones, Keith van Rijsbergen y Ricardo Baeza-Yates.

A1.1.1 Definiciones.

Los sistemas de búsqueda y recuperación de información tienen objetivos diferentes:

Recuperación de Información *(Information Retrieval)*:

Según define el profesor Lluís Codina [COD95], la recuperación de información es una operación que consiste en la interpretación de una necesidad de información con el fin de seleccionar los documentos más relevantes capaces de solucionarla.

Ejemplo de estos sistemas son los buscadores de Internet (Google[11], Yahoo[12], Bing[13] etc.) que, a partir de una serie de palabras clave, operadores booleanos y otros criterios como la lengua, la fecha de publicación, el país del dominio y demás, devuelven una lista de documentos ordenados según la relevancia que el sistema considera que pueden tener, en función de parámetros como el número de apariciones de las palabras clave o enlaces que apuntan al sitio Web.

El principal inconveniente de estos sistemas es el hecho de que sea el usuario el que debe ir abriendo documentos para encontrar la información que realmente le interesa.

Extracción de Información *(Information Extraction):*

Sistemas que procesan colecciones de documentos de un dominio dado para extraer la información relevante de manera estructurada. Son sistemas hechos a medida y normalmente reciben una plantilla definida que deben rellenar con información extraída del texto. S. Soderland [SOD99] lo define como: “un sistema de extracción de información puede servir como una interfaz para una recuperación de información de gran precisión, como un primer paso en sistemas de búsqueda en grandes cantidades de textos, o como entrada a un agente inteligente cuyas acciones dependen de la comprensión del contenido de una información almacenada en forma de textos”.

Un ejemplo de este tipo podría ser nuestro sistema que, teniendo como entrada las noticias de un diario, las relaciona y obtiene datos biográficos de un político concreto entre unas fechas dadas. A partir de esta información, como trabajo futuro, se podría generar una base de datos estructurada (es lo que se denomina Integración de la Información) para agilizar las posteriores consultas sobre los actos de la persona.

10 http://www.itl.nist.gov/iaui/894.02/related_projects/tipster/
11 http://www.google.com
12 http://www.yahoo.com
13 http://www.bing.com

El principal inconveniente de estos sistemas es la poca portabilidad que demuestran ya que las plantillas que utilizan están diseñadas para un fin concreto y son difícilmente reutilizables. Así que tendremos que intentar que nuestro sistema sea lo más flexible posible (ficheros de configuración, otros ficheros externos para hacernos independientes de cambios en los recursos, etc.)

Búsqueda de información:

Un proceso de recuperación de información comienza cuando un usuario hace una consulta al sistema. Una consulta a su vez es una afirmación formal de la necesidad de una información. En la recuperación de información una consulta no identifica únicamente a un objeto dentro de la colección. De hecho varios objetos pueden ser respuesta a una consulta con diferentes grados de relevancia.

Un objeto es una identidad que está representada por información en una base de datos. Dependiendo de la aplicación estos objetos pueden ser archivos de texto, imágenes, audio, mapas, videos, etc. Muy a menudo los documentos no están almacenados en el sistema de recuperación de información, sino que están referenciados.

A1.1.2 Categorización de los modelos de recuperación de información.

La mayoría de los sistemas de recuperación de información computan un ranking para saber cómo cada objeto responde a la consulta, ordenando los objetos de acuerdo a su valor de ranking. Los objetos con mayor ranking son mostrados a los usuarios y el proceso puede tener otras iteraciones si el usuario desea refinar su consulta.

Para recuperar efectivamente los documentos relevantes por medio de estrategias de recuperación de información, los documentos son transformados en una representación lógica de los mismos. Cada estrategia de recuperación incorpora un modelo específico para sus propósitos de representación de los documentos. La Figura 2.1 ilustra la relación entre algunos de los modelos más comunes. Los modelos están categorizados de acuerdo a dos dimensiones: la base matemática y las propiedades de los modelos.

Primera Dimensión: Base matemática

- Modelos basados en Teoría de Conjuntos: Los documentos se representan como un conjunto de palabras o frases. Los modelos más comunes son:
    - Modelo Booleano
    - Modelo Booleano Extendido
    - Modelo Difuso

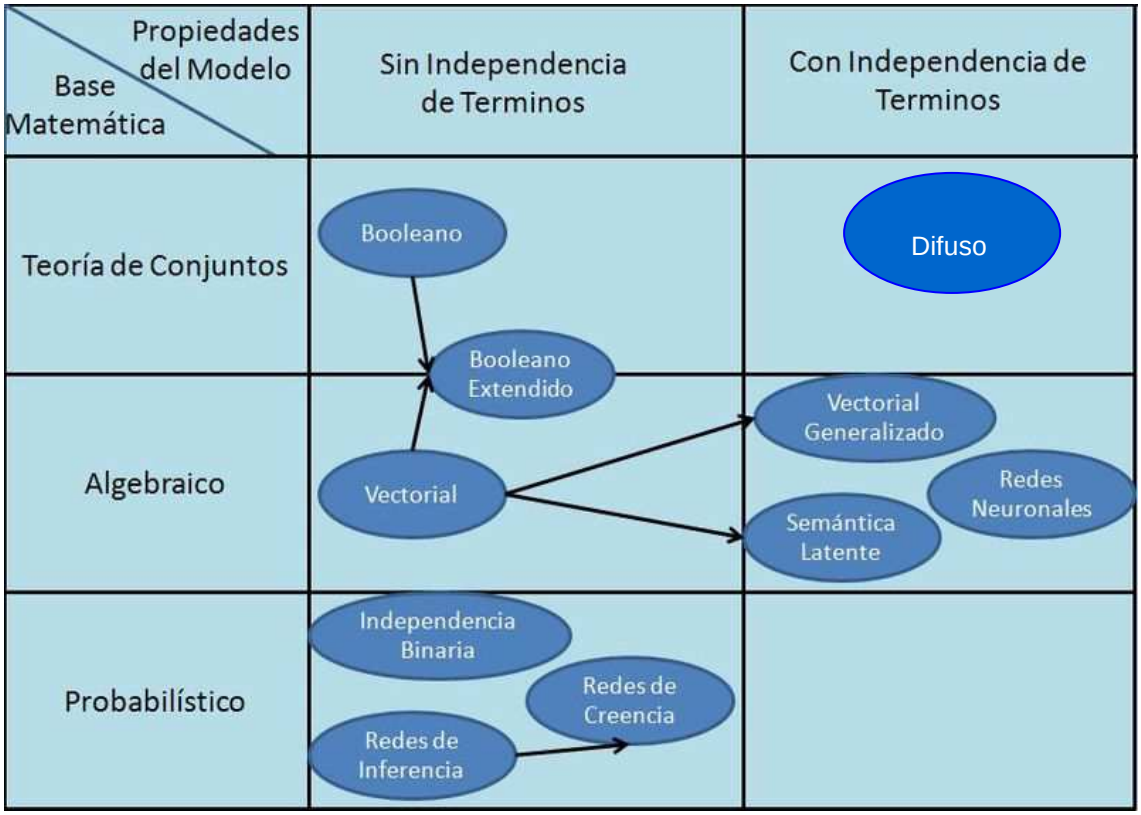

Fig. 2.1 Categorización de los Modelos de Recuperación de Información

- Modelos Algebraicos: En estos modelos los documentos y las consultas se representas como vectores, matrices o tuplas. Codina [COD95] nos habla del modelo vectorial: "En el modelo de comparación que está basado en la utilización de espacios vectoriales de *n* dimensiones, desarrollado principalmente por Salton (1983) (puesto que los documentos se representan como vectores) los documentos pueden situarse en un espacio vectorial de *n* dimensiones, es decir, con tantas dimensiones como elementos tenga el vector. A nosotros nos cuesta imaginar un espacio vectorial de más de tres dimensiones, pero matemáticamente resulta rutinario tratarlos.

  Situado en ese espacio vectorial, cada documento *encaja* entonces en un lugar determinado por sus coordenadas, al igual que en un espacio de tres dimensiones cada objeto queda bien ubicado si especificamos sus tres coordenadas espaciales.

  Se crean así grupos de documentos que quedan próximos entre sí a causa de las características de sus vectores. Estos grupos o *clusters* están formados, en teoría, por documentos similares, es decir, "por grupos de documentos que son relevantes para la misma clase de problemas de información."

Dentro del modelo vectorial se distinguen:

- Modelo Vectorial
- Modelo Vectorial Generalizado
- Modelo Booleano Extendido
- Indexación Semántica Latente

- Modelos Probabilísticos: Tratan el proceso de recuperación de documentos como una inferencia probabilística. Las similitudes son calculadas como las porobabilidades de que un documento sea relevante dada una consulta.
  - Modelo de independencia binaria
  - Modelo de Relevancia Probabilístico
  - Redes de Inferencia
  - Redes de Creencia

Segunda Dimensión: Propiedades de los Modelos

- Modelos sin independencia entre términos: Tratan a los términos como si fueran independientes.
- Modelos con dependencia entre términos: Permiten representar las interdependencias entre términos.

Aplicaciones actuales de la teoría de RI

Aparte de su posible elegancia intelectual, ¿qué aplicaciones tienen estas teorías y formalismos? Por lo pronto, cada vez más sistemas de gestión documental como los citados al principio han decidido incorporar, de una manera más o menos transparente, el cálculo de relevancia al preparar los documentos para presentarlos al usuario, en lugar de presentar los documentos aleatoriamente.

En general, la idea que subyace tras estas ordenaciones es más sutil e importante de lo que parece si se examina con cuidado. Cuando un usuario formula una pregunta a una base de datos documental espera recuperar una cantidad *n* de documentos que satisfagan su necesidad de información. Pero el valor de *n* es desconocido, pueden ser dos o tres documentos, o pueden ser miles.

La situación habitual ante una gran cantidad de documentos recuperados consiste en utilizar nuevos términos de búsqueda combinados por operadores booleanos, para ir restringiendo el número total, y dejarlo así por debajo del *punto de inutilidad* [BLAIR90], es decir, por debajo de aquella cantidad de información que el usuario preferirá no leer, dado su volumen.

Sin embargo, para que la operación anterior tenga éxito, debe tenerse un buen conocimiento de la base de datos, un buen dominio de la lógica booleana y un buen conocimiento del tema de búsqueda, además de tiempo y paciencia y, al final, puede que las operaciones booleanas reduzcan el tamaño del conjunto recuperado a costa de perder documentos relevantes [BLAIR90].

En cambio, con la ordenación por grado de relevancia, no importa lo grande que sea el número de documentos recuperados; el usuario sabe que justo los primeros son los

más relevantes y, por tanto, le bastará con atender sólo a aquellos documentos que hayan superado un cierto umbral de relevancia, y como el primer documento es el más pertinente, el segundo lo es sólo un poco menos, etc. Él puede situar el umbral donde crea conveniente, según el tiempo de que disponga, el tamaño de los documentos, la complejidad del tema, etc.

Para localizar los artículos más relevantes, he optado por profundizar en el uso del algoritmo TF-IDF, ya que, tal como se cita en el artículo de Carrera, Cisne y Pasmay [CPC12], es uno de los más utilizados en motores de búsqueda.

A1.1.3 Algoritmo TF_IDF

El coeficiente TFIDF *Term Fre-quency–Inverse Document Frequency* [SAMCG86] es una ponderación usada a menudo en tareas de recuperación de información y minería de texto. El coeficiente es una medida estadística usada para evaluar cómo de importante es una palabra respecto a un documento perteneciente a una colección o cuerpo de documentos. La importancia de cada palabra incrementa proporcionalmente con el número de veces que ella aparece en el documento pero se ve influenciada por la frecuencia de la palabra en el cuerpo de documentos. Variaciones del esquema de ponderación TFIDF son usados a menudo por los motores de búsqueda como una herramienta para la puntuación y creación de ranking de la relevancia de un documento ante una consulta de usuario determinada.

Una de las funciones de ranking más sencillas se calcula como la suma de los valores tf-idf de cada término de la consulta. Muchas funciones de ranking más complejas constituyen variaciones de este simple modelo.

Supongamos que tenemos una colección de documentos y queremos determinar el documento más relevante a la consulta "la mochila azul". Una manera sencilla de comenzar es eliminando aquellos documentos que no contengan las tres palabras "la", "mochila" y "azul", pero todavía quedan muchos documentos. Para diferenciarlos aún más, debemos contar el número de veces que cada término ocurre en cada documento y sumarlos; el número de veces que un término ocurre en un documento se denomina su frecuencia de término (tf).

Sin embargo, como el término "la" es tan común, esto provocará que se destaquen incorrectamente documentos que utilizan de casualidad la palabra "la" con más frecuencia, sin conceder suficiente peso a los términos más significativos "mochila" y "azul". El término "la" no es una buena palabra clave para distinguir documentos relevantes y no relevantes, a diferencia de las palabras menos comunes "mochila" y "azul". Por lo tanto, se incorpora un factor de frecuencia inversa de documento que atenúa el peso de los términos que ocurren con mucha frecuencia en la colección de documentos e incrementa el peso de los términos que ocurren pocas veces.

Tf-idf es el producto de dos medidas, frecuencia de término y frecuencia inversa de documento.

La frecuencia de un término (TF) en un documento dado es simplemente el número de veces que el término aparece en ese documento. Este valor es usualmente normalizado para prevenir que documentos extensos adquieran una inusual ventaja. De esta forma, la importancia del término t_i en el documento d_j está dada por:

$$TF_{i,j} = \frac{n_{i,j}}{\sum_k n_{k,j}}$$

donde $n_{i;j}$ es el número de ocurrencias de término considerado en el documento d_j , y el denominador es el número de ocurrencias de todos los términos en el documento d_j .
La frecuencia inversa de los documentos (IDF) es una medida de la importancia general del término y se calcula mediante:

$$IDF_i = \log \frac{|D|}{|\{d_j : t_i \in d_j\}|}$$

donde el numerador es el número total de documentos en el cuerpo y el denominador es el número de documentos donde el término t_i aparece (i.e., $n_{i;j} \neq 0$).

Así, el coeficiente TFIDF para el término t_i en el documento d_j es:

$TFIDF_{i;j} = TF_{i;j} \times IDF_i$

Un valor TFIDF alto es alcanzado por un término con alta frecuencia en el documento considerado, pero baja frecuencia en la colección total de documentos. De esta manera, el coeficiente tiende a filtrar términos comunes.

A1.2 Extracción de información

El estudio de esta área surge de la necesidad de extraer datos concretos que están embebidos en textos no estructurados como son los artículos periodísticos o enciclopédicos. El generador de fichas de biografías necesita presentar fechas o localidades de nacimiento, muerte, obras, etc., que seguramente aparecen en los artículos que se han ido escribiendo sobre ese personaje, pero que nuestra herramienta debe saber localizar y extraer.

A1.2.1 Introducción

El tesoro más valioso de la raza humana es el conocimiento, es decir, la información. Existen en el mundo volúmenes inmensos de información en forma de lenguaje natural: los libros, los periódicos, los informes técnicos, etc. Pero la posesión verdadera de este tesoro implica la habilidad de hacer ciertas operaciones con la información:

- Buscar la información necesaria.
- Comparar las fuentes diferentes, y hacer inferencias lógicas y conclusiones.
- Manejar los textos, por ejemplo, traducirlos a otros idiomas.

En realidad, las computadoras son más capaces de procesar la información que las personas. Pueden procesar muchísimos más volúmenes de información de los que una persona puede leer en su vida. Y basándose en esta información, pueden hacer inferencias lógicas tomando en cuenta más hechos y más fuentes.

Se denomina Procesamiento del Lenguaje Natural (PLN, del inglés *Natural Language*

Processing, NLP) a la rama que surge de la Inteligencia Artificial y la Lingüística Computacional y que se ocupa de investigar y formular mecanismos eficaces para la comunicación entre personas o entre personas y máquinas por medio de lenguajes naturales y que además se puedan realizar por medio de programas que ejecuten o simulen esta comunicación.

El procesamiento del lenguaje natural presenta múltiples aplicaciones que se citan en la introducción del libro de Manning [MAN99]:

- Corrección de textos
- Traducción automática
- Recuperación de la información
- Extracción de Información y Resúmenes
- Búsqueda de documentos
- Sistemas Inteligentes para la Educación y el Entrenamiento

La corrección de textos permite la detección y corrección de errores ortográficos y gramaticales. Para detectar este tipo de errores, la computadora necesita entender en cierto grado el sentido del texto. Los correctores de gramática detectan las estructuras incorrectas en las oraciones aunque todas las palabras en la oración estén bien escritas en el lenguaje en cuestión. El problema de detectar los errores de este tipo es complejo debido a la existencia de gran variedad de estructuras permitidas.

Para describir las estructuras de las oraciones en el idioma, se usan las llamadas gramáticas formales, o sea conjuntos de reglas de combinación de palabras y su orden relativo en las oraciones.

La traducción automática se refiere a la traducción correcta de un lenguaje a otro, tomando en cuenta lo que se quiere expresar en cada oración.

En el campo de la recuperación de la información han desarrollado sistemas que permiten obtener información sobre estadísticas deportivas, información turística, geografía etc. En lugar de buscar los documentos para encontrar en ellos la respuesta a su pregunta, el usuario podría hacer su pregunta a la computadora: ¿Cómo se llama el Presidente de Francia?, ¿Cuáles son los centros más avanzados en Procesamiento del Lenguaje Natural?, y otras. El modelo de recuperación de información que utiliza técnicas de PLN para responder preguntas formuladas en lenguaje natural se llama *Búsqueda de respuestas (Question Answering, QA)*.

Las técnicas de QA suponen un paso más en la extracción de información, y como nos dice el artículo de Ravichandran y Hovy [DEHO02] se pueden crear patrones que encontramos en el texto como respuestas a determinadas preguntas y de esta manera, conseguir que el sistema aprenda a responder a dichas preguntas. De hecho, ya existen en el mercado buscadores que ofrecen una interacción QA con el usuario, como es answerbase.com[14]. El primer buscador de este tipo, MULDER, se describe en el artículo de Kwok, Etzioni y Weld “Scaling Question Answering to the Web” [COETWE01].

Por otra parte se han desarrollado sistemas con la capacidad de crear resúmenes de documentos a partir de los datos suministrados. Estos sistemas son capaces de realizar un análisis detallado del contenido del texto y elaborar un resumen. Como ejemplos, se pueden ver los *papers* sobre generación de resúmenes de [LLOPA11], [PLADI11] y [BOSAGG11] en los que se usan diferentes técnicas para elaborar resúmenes de textos.

Uno de los elementos fundamentales en el diseño de un sistema PLN es sin lugar a dudas la determinación de la arquitectura del sistema, es decir, cómo se introducen los

[14] http://answerbase.com/

datos a la computadora y cómo ella interpreta y analiza las oraciones que le sean proporcionadas. A continuación se muestra un esquema del análisis léxico/ sintáctico por computadora [Grish97]. El sistema consiste de:

a. El usuario le expresa (de alguna forma) a la computadora qué tipo de procesamiento desea hacer;
b. La computadora analiza las oraciones proporcionadas, en el sentido morfológico y sintáctico;
c. Luego, se analizan las oraciones semánticamente, es decir se determina el significado de cada oración;
d. Se realiza el análisis pragmático del texto. Así, se obtiene una expresión final.

El Procesamiento del Lenguaje Natural (PLN) es una de las piedras angulares que primero surgen de la inteligencia artificial (IA). La Traducción Automática, por ejemplo, nació a finales de la década de los cuarenta, antes de que se acuñara la propia expresión «Inteligencia Artificial». No obstante, el PLN ha desempeñado múltiples papeles en el contexto de la IA, y su importancia dentro de este campo ha crecido y decrecido a consecuencia de cambios tecnológicos y científicos. Los primeros intentos de traducir textos por ordenador a finales de los cuarenta y durante los cincuenta fracasaron debido a la escasa potencia de los ordenadores y a la escasa sofisticación lingüística. Sin embargo, los esfuerzos realizados en las décadas de los sesenta y los setenta para producir interfaces en lenguaje natural para bases de datos y otras aplicaciones informáticas obtuvieron un cierto grado significativo de éxito. La década de los ochenta y el principio de la de los noventa han visto resurgir la investigación en el terreno de la Traducción Automática.

Actualmente (se nos explica en el libro de Bird y Klein *Natural language processing with Python* [BIRKLE09]) las tecnologías basadas en NLP están cada vez más extendidas. Por ejemplo, los teléfonos móviles y tabletas soportan texto predictivo y reconocen la escritura hecha a mano; los motores de búsqueda dan acceso a información encerrada en textos sin estructura; la traducción automática nos permite utilizar textos escritos en otros idiomas y traducidos al nuestro. *Python* es un lenguaje que actualmente sirve para crear interfaces basados en lenguaje natural entre hombres y máquinas, y nos proporciona un acceso más sofisticado a la información.

A1.2.2 Minería de textos

Teniendo en cuenta que extraer la información manualmente requería un gran esfuerzo humano tanto de tiempo como de recursos, la comunidad científica dedicada al Procesamiento de Lenguaje Natural (PLN) mostró su interés por esta problemática y empezó a desarrollar técnicas para resolverla.

Algunos de los problemas a la hora de recuperar información provocados por el uso del lenguaje natural (entre otras razones) son el silencio (debido a la sinonimia), el ruido (debido a la polisemia), homografía, ambigüedad, etc.

La minería de textos o *Text Mining* surge a raíz de estas necesidades. Es una de las ramas de la lingüística computacional que trata de obtener información y conocimiento a partir de conjuntos de datos que en principio no tienen un orden o no están dispuestos en origen para transmitir esa información.

Hoy día, se le presta cada vez más atención a la minería de textos multilingüe debido a la proliferación de información multilingüe en Internet y la conveniencia de no

restringir el rango de la búsqueda a documentos en la misma lengua en que se formula la consulta.

La minería de textos [CAR97] tiene como objetivo tratar grandes colecciones documentales y extraer determinada información de ellas. Su origen se halla a principios de los años ochenta, tras constatar que:

- La mayor parte de la información se encuentra de manera textual,
- Dicha información reside en soportes digitales y,
- Debido al gran valor comercial que posee, interesa su extracción y clasificación.

Centrándonos más en el tipo de obtención de datos que requiere nuestro proyecto, encontramos que la extracción de información biográfica tiene como objetivo crear de manera automática grandes repositorios que contengan información semántica estructurada acerca de personajes públicos: ocupación, fecha y lugar de nacimiento y/o muerte, obra, etc.

Esta información puede ser utilizada posteriormente en sistemas de búsqueda de respuestas, en procesos de recuperación de información o en ampliación de ontologías, por ejemplo.

Existen dos estrategias fundamentales en la extracción de información de fuentes no estructuradas: una consiste en obtener oraciones que contengan pares de entidades potencialmente representativos de una relación semántica (NombreDePersona, FechaDeNacimiento); otra consiste en decidir si realmente el par contiene esa relación en la oración extraída.

Es habitual en la extracción de información el encontrar el problema que presentan los nombres de persona (NPer) que pueden presentarse de varias maneras: “Lorca” (que a su vez puede referirse a una localidad), “Federico García Lorca”, “García Lorca”, etc. Y además este nombre puede aparecer referido a una calle, un centro cultural, etc…

A1.2.3 Named Entities

La búsqueda en la Web nos proporciona un vasto conjunto de enlaces a los documentos seleccionados entre billones de los disponibles. La responsabilidad de pasar de la totalidad de los documentos a encontrar la información buscada (normalmente un conjunto de frases o párrafos relevantes) recae en los usuarios, que deben buscar entre los documentos que devuelve la búsqueda para encontrar el que realmente contiene los datos requeridos.

Un caso frecuente, son las consultas sobre *Named Entities* (nombres propios, en adelante NE) que forman una parte importante de las entradas en motores de búsqueda. Cuando se envían consultas como John Williams o Phyton, los motores de búsqueda pueden devolver una cantidad de datos o atributos sobre estas NE, como el conjunto de las páginas web con la mayor coincidencia en esos nombres.
Uno de los retos al crear búsquedas alternativas es la ambigüedad de las consultas, como varias instancias de la misma clase (como diferentes personas) o diferentes clases (por ejemplo, un tipo de serpiente, un lenguaje de programación, o una película) pueden compartir el mismo nombre en la consulta. En el caso de Python por ejemplo, en cualquier buscador encontramos entradas que se refieren al lenguaje de programación, la serpiente o los *Monty Python*.

Parece natural tratar de explotar la web para mejorar el funcionamiento de la extracción de relaciones, como por ejemplo, descubrir relaciones útiles entre las NE mencionadas en documentos de texto. En el *paper* [NAGHAS98] tenemos un algoritmo para identificar NE, resolver anáforas y correferencias, y utilizar todo esto para localizar frases relevantes de un texto. En [CAMARPA03] se describen otras técnicas para identificar y clasificar las NE.

Sin embargo, si queremos combinar información de múltiples páginas web, necesitamos resolver el problema de la desambiguación. Si no lo hacemos, el sistema que extraiga las relaciones y que esté analizando frases de varios documentos puede considerar por error que una NE es la autora de un hecho que no le corresponde. Por ejemplo, confundir películas que ha interpretado Jorge García, actor estadounidense, y asignárselas a Jorge García, futbolista del Sporting de Gijón.

El término *Named Entity* (NE) fue introducido en Noviembre de 1995 en la MUC-6[15].
MUC-6 fue la sexta Conferencia en una serie llamada "Conferencias para comprensión de los mensajes" (*Message Understanding Conferences)*. Se celebró en Noviembre de 1995, y como las anteriores fue organizada por Beth Sundheim del grupo de Investigación y Desarrollo Naval. Todas las conferencias se dedicaron a la evaluación de sistemas de extracción de información. Las cinco previas se habían centrado en la única tarea de extraer información: analizar el texto, etiquetar y almacenar las etiquetas en plantillas de almacenamiento. En esta última dirigida por Ralph Grishman, se definen una serie de objetivos para que las técnicas de extracción de información fueran trasladables a más dominios de conocimiento, y que se pudiera trabajar en el lenguage natural. Así desarrollaron definiciones para las siguientes tareas:

-Reconocimiento de *Named Entities,*
-Correferencias,
-Elementos para las plantillas
-Escenarios de las plantillas

Estos dos últimos puntos tratan de definir tipos de eventos ("escenarios") o clases, y las plantillas contendrían los datos que definen cada instancia de cada clase.

En esta conferencia, aparte de la evaluación de los sistemas habitual, se abarcó la importancia del reconocimiento de la información. Los sustantivos que se refieren a entidades individuales (nombres propios de persona, nombres de organizaciones y nombres locativos entre otros) junto con las expresiones numéricas (fechas, tiempo, porcentajes, dinero) destacaban en esta área. A partir de las conclusiones obtenidas y dada su importancia, las NE fueron ampliadas con la finalidad de poder adaptarse a los diferentes ámbitos de trabajo (biología, política, medicina, tecnología) dando lugar a las Extended Named Entities.

De todas formas, la clasificación de NE más utilizada continúa siendo la establecida por MUC:

-Nombres de Persona.
-Organizaciones.
-Nombres locativos (políticos o físicos).
-Expresiones numéricas fecha-tiempo.
-Otras NE (medidas -porcentajes, monetarias, pesos-, direcciones de correo, direcciones web, etc.).

[15] http://cs.nyu.edu/cs/faculty/grishman/muc6.html

A1.2.4 Resolución de correferencias

La correferencia es un concepto gramatical que indica igualdad (de ahí el prefijo co-) en la referencia de dos o más elementos lingüísticos.

La resolución de correferencias entre las diferentes variantes de nombre de persona y la desambiguación son fundamentales en nuestro proyecto para la obtención y estructuración de información.

La popularización de los sistemas de reconocimiento y clasificación de NE (NER/NEC: *Named Entities Recognition / Named Entities Clasification)* ha resuelto gran parte del trabajo previo de detección y etiquetado de nombres propios, asi como la detección de las fronteras de los nombres propios, y la resolución de ambigüedades semánticas: distinción entre personas, localidades, organizaciones u otras. En este caso, la herramienta que utilizaremos es Freeling, paquete software analizador de idiomas de código libre desarrollado por TALP[16] de la UPC[17]. Este sistema es capaz de reconocer y clasificar las NE de lengua castellana gracias a las reglas integradas que posee del idioma. En inglés y catalán, FreeLing incluye NER pero no NEC.

Las aproximaciones a la resolución total de correferencia (no sólo entre nombres propios o de persona) han sido objeto de varias evaluaciones (MUC-7 o SemEval 2010). Dada su complejidad, este tipo de tareas requieren un mayor número de recursos, tales como corpus anotados, analizadores sintácticos, etc.

Existen dos tipos fundamentales de aproximación a la resolución de correferencias entre nombres de persona, en función de los objetivos o de los recursos utilizados. El primero, correferencia con foco, se aplica sobre textos cuyo foco se centra en una única persona, como son los artículos en enciclopedias o biográficos. El segundo, correferencia completa, trata de establecer las relaciones entre todos los nombres de persona que compartan algún referente en un texto.

También es importante tener en cuenta que en la detección y resolución de correferencias entre nombres de persona entran en juego varios factores: además del propio algoritmo de resolución, este puede requerir recursos externos de conocimiento (listas de variantes, keywords, equivalencias, etc.) o sistemas de reconocimiento y clasificación de nombres propios. También hay que tener en cuenta los tipos de correferencias posibles: Inclusión (por ejemplo, “Lennon” y “John Lennon”), abreviatura (“John Fitzgerald Kennedy” y “JFK”), hipocorísticos y diminutivos (“José Blanco” y “Pepe Blanco”, o “Pepiño”), apodos (“Ronaldo de Assis Moreira” y “Ronaldinho”), Foco / Conocimiento General (incluir “Bush” en el mismo artículo donde se nombra a su esposa “Laura Bush”).

A1.2.5 Desambiguación

Desambiguar es efectuar las operaciones necesarias para que una palabra, frase o texto pierdan su ambigüedad. Y tal como nos cita Wikipedia, en lingüística computacional, la desambiguación del significado de la palabra es un problema abierto

[16] Centre de Tecnologies i Aplicacions del Llenguatge y la Parla

17 Universitat Politècnica de Catalunya

de procesamiento de lenguaje natural, que incluye el proceso de identificar qué sentido de palabra está usada en los términos de una oración, cuando la palabra en cuestión tiene polisemia, es decir, pluralidad de significados.

En el *paper* "*Word Sense Disambiguation: A Survey"* [NAV09] se indica que la tarea de desambiguación del significado de la palabra (en el original WSD *Word Sense Disambiguation)* es histórica en el campo del procesamiento del lenguaje natural. Define esta tarea como la capacidad de determinar a través de un proceso computacional cuál es el sentido o significado de una palabra dependiendo del contexto. WSD puede entenderse así como una tarea de clasificación: los significados de una palabra son las clases, y se utiliza un método de clasificación automático para asignar a cada instancia de una palabra una o más clases dependiendo del contexto y de otras fuentes externas que nos aporten datos.

Existen diferentes alternativas para la desambiguación de *Named Entities*, en este caso hemos estudiado dos artículos con visiones distintas. La primera visión es interesante, porque se basa en un diccionario construido a partir de Wikipedia, y la construcción de un diccionario va a ser necesaria para nuestro proyecto.

Primera solución: utilizar un diccionario.

En el artículo de Bunescu y Pasca [BUPA06] presentan un método para desambiguar NE ligado a un diccionario: en primer lugar detecta cuándo un nombre propio se refiere a una NE incluida en el diccionario (detección), y en segundo lugar, realiza la desambiguación entre múltiples NE que puedan ser referidas por el mismo nombre propio (desambiguación).
Para realizar estas funciones utilizan Wikipedia de la que utilizará como recursos sus "redirect page" (existe una para cada nombre alternativo que se puede usar para referirse a una entidad en Wikipedia), y las "disambiguation pages", creadas para nombres que denotan dos o mas entidades en Wikipedia.

Para la desambiguación, utilizarán también un diccionario que crean con todas las NE de Wikipedia (En nuestro caso, lo realizaremos a partir de un catálogo de *Named Entities* creado previamente, y apoyándonos en el tesauro del archivo documental).

El procedimiento según Brunescu y Pasca para la creación del diccionario es el siguiente: Como cada título en Wikipedia debe empezar por mayúsculas, la decisión de cuándo un título es un nombre propio debe recaer en las heurísticas que se definen a continuación.

1. Si *título* es un título con múltiples palabras, se comprueba cuántas de ellas comienzan por mayúscula y son distintas de preposiciones, determinantes, conjunciones, pronombres relativos o negaciones (las llamaremos palabras con significado). Consideraremos que hemos encontrado una NE si y sólo si todas la palabras con significado comienzan por mayúscula.

2. Si *título* es un título de una sola palabra y contiene al menos dos letras mayúsculas, entonces se considera esa palabra como NE. En otro caso, pasamos al punto 3.

3. Contamos cuantas veces aparece *título* en el texto del artículo, es posiciones distintas al comienzo de las frases. Si al menos el 75% de las apariciones lo hace con mayúscula, entonces consideraremos esa palabra como NE.

Estas heurísticas combinadas extraen cerca de millón y medio de NE de la Wikipedia. En nuestro caso, contamos con Freeling y el tesauro para ayudarnos a extraer las *Named Entities*. El segundo paso es construir el diccionario.

- El conjunto de entradas del diccionario consiste en todas las cadenas que pueden denotar una NE. Por ejemplo, si *título* es una NE, entonces se añade como entrada al diccionario el mismo *título*, sus "*redirect names*" y sus "*disambiguation names*" de las páginas de Wikipedia mencionadas anteriormente.
- Cada entrada se mapea con el conjunto de NE extraídas anteriormente, de manera que una NE se incluye en el diccionario si ésta coincide en el título, o pertenece al conjunto de "*redirect names*" o "*disambiguation names*".

Cuando se realiza una consulta sobre un nombre propio, consultaremos al diccionario. Si existe una entrada que coincide con el nombre propio y ésta contiene al menos dos entidades, una de ellas será la correcta.

Para la desambiguación utilizan dos algoritmos. El primero utiliza la similitud en contexto de los artículos. Este algoritmo resuelve la desambiguación de NE como un problema de ranking, utilizando funciones de puntuación para escoger la opción más puntuada utilizando la frecuencia de la aparición de la palabra en un documento en el conjunto de documentos de Wikipedia. Un segundo algoritmo corrige errores debidos a artículos demasiado cortos o incompletos, y otros artículos que se refieren al mismo concepto pero empleando sinónimos.

Segunda solución: sistemas basados en reglas.

Un segundo artículo de Garcia y Gamallo, "Resolución de Correferencia de Nombres de Persona para Extracción de Información Biográfica" [GARGA11], propone una solución que nos parece interesante por no necesitar un corpus de entrenamiento previo, y propone unas herramientas y recursos externos que no son difíciles de conseguir.

La propuesta consiste en lo siguiente:

En él proponen dos algoritmos junto a un *baseline*. Estos sistemas están basados en reglas, por lo que no necesitan un corpus de entrenamiento previamente etiquetado, y son independientes de una lengua particular. Sólo los recursos externos son dependientes del idioma, pero se pueden extraer fácilmente de manera automática.

Como *baseline* se han extraído de Wikipedia los títulos de artículos pertenecientes a categorías relativas a personas. De todos los títulos, se han escogido solamente la forma enciclopédica (por ejemplo, de las formas "John Lennon", "Lennon", "John Winston Ono Lennon" o "Discografía de John Lennon", únicamente la primera será extraída).

El primer algoritmo utiliza una estrategia "Partial-match". Para aplicarlo serán necesarias como herramientas un tokenizador, un NER (*Named Entities Recognition)*, y un NEC (*Named Entities Classification* que etiquete las entidades como persona, organización, localidad u otros).

a) ALGORITMO “PARTIAL-MATCH”

Para cada uno de los nombres etiquetados como “persona” por el NEC:

1. Establece relaciones entre los nombres de persona (en adelante NPer) con la misma forma
2. Cada NPer es dividido en tokens, ignorando las palabras vacías que puedan contener (<de>, <la>, etc.)
3. Los tokens de un NPer (A) son comparados con los tokens de otros (B), estableciéndose una relación de correferencia si algún token de A existe en B, excepto:

-En casos en que un nombre simple (“John”) pueda coincidir con parte de más de un nombre compuesto (“John Lennon” o “John F. Kennedy”), donde la relación se establece entre el nombre compuesto anterior más próximo.

-En casos en los que el número de coincidencias entre los tokens de A y B es menor que entre B y C: (A) “Fernando González Ochoa”; (B) “José González” y (C) “José González Torres”.

B) ALGORITMO NP-VAR

Para aplicar el siguiente algoritmo es necesario obtener (además de un conjunto de palabras vacías) las siguientes listas de palabras:

Trigger-words de Persona (tw-p): palabras que indican que la entidad es una persona (presidente, cónsul, etc.). Son obtenidas automáticamente a través de métodos utilizados en el desarrollo de herramientas NEC, o extraídos directamente de recursos libres.

Trigger-words de Localidades, Organizaciones (y otras) (tw-block): calle, museo, instituto, gobierno, obra, etc. Obtenidas también de sistemas NEC libres y aumentadas automáticamente de la siguiente manera: se extraen todos los nombres propios de un corpus de gran tamaño (en este caso, la Wikipedia), y se seleccionan aquellos cuyo primer token sea una palabra de diccionario (por ejemplo, “museo”) y su segundo token sea un NPer (estos últimos previamente extraídos de modo automático de la Wikipedia). La lista es posteriormente filtrada con nombres de pila de persona, para evitar entradas que, siendo formas de diccionario, sean también nombres de pila (por ejemplo, “Rosa”).

Una vez obtenidos estos recursos, para aplicar el algoritmo, el texto fuente es también analizado con un tokenizador, un NER y un NEC.

El primer paso consiste en decidir si el documento analizado tiene como foco un NPer. Si el título o encabezamiento del documento se corresponde con un NPer (p. e., artículo enciclopédico), este es seleccionado como foco. Si no,

i. se selecciona el NPer de un solo token más frecuente del documento;
ii. se verifica si este nombre forma parte de uno de los n NPer compuestos más frecuentes del texto (donde n ha sido definido empíricamente como 2);
iii. si es así, se define el nombre compuesto más frecuente como foco, salvo que exista otro NPer compuesto de frecuencia similar (>= 0, 6) que también contenga el nombre simple. El grado de similaridad entre el primer y el segundo candidato ha sido establecido por los autores del artículo en 0,6 después de haber probado diferentes valores.

Si el documento tiene como foco un NPer, todos las apariciones simples de cada uno de los tokens del nombre-foco se etiquetan como correferencial de este.

Tanto si se ha encontrado foco como si no, se busca una coincidencia de patrones entre los NPer y NPmisc, estableciendo correferencias entre ellos. Hay que tener en cuenta que el algoritmo utiliza también las entidades misc (que no son persona, localidad u organización), debido a que pueden corresponderse con NPer mal etiquetados.

A continuación, cada NPer y NPmisc encontrado desde el inicio (A), es comparado con los anteriores (B):

1. Si A sólo tiene un token, se considera correferencial de B si B contiene ese token (y A no es una *tw-block*).

2. Si A tiene más de un token, se considera correferencial de B si B contiene a A (ignorando las palabras vacías y las tw-p), salvo que A o B terminen en una keyword negativa (“Jr.”), no presente en el otro nombre.

3. Si A tiene un token de un solo carácter (o una letra y un punto), que no sea el último, se selecciona el primer carácter de todos los tokens de A y B (salvo el último, las palabras vacías y tw-p), y se comparan: si B contiene a A, se consideran correferenciales.

4. Si el token de un único carácter es el último, se realiza la misma operación, pero incluyendo todos los tokens (excepto palabras vacías y tw-p).

5. Después de recorrer todos los nombres del documento, el algoritmo vuelve al principio. Todos aquellos NPer y NPmisc para los que no ha sido encontrado un correferente son comparados con el NPer y NPmisc siguiente más próximo, realizando las mismas operaciones (nótese que ahora A y B se invierten).

Finalizada la ejecución, se habrán obtenido todas las variantes de cada uno de los nombres propios, y su lema es substituido por una forma canónica (el nombre más largo y una identificación numérica).

En el artículo en el que se presentan estos algoritmos, aparece la evaluación MUC propuesta en 1995, dando los siguientes resultados de precisión (P), recall (R) (links correctos sobre los existentes), y f-score (F). Nos parece interesante tenerla en cuenta, para valorar decisiones posteriores a la hora de implementarlos o no.

El corpus de evaluación está formado por textos enciclopédicos (extraidos de Wikipedia, de la categoría “Escritores en español”), y periodísticos (noticias tomadas del periódico “El País”).

Positiva		**Wikip.**	**El País**	**Total**
Basel.	P	100	100	100
	R	27,7	29,1	28,2
	F	43,4	45,1	**44**
P-Match	P	79,4	95,6	86
	R	89,9	93,6	91,2
	F	84,3	94,6	**88,5**
NP-Var	P	94,5	98,4	96
	R	89,9	91,4	90,5
	F	92,2	94,8	**93,1**

Tabla 2.1: Resultados de precisión, recall y f-score extraídos de [GARGA11]

A2. RECONOCIMIENTO FACIAL

En este capítulo se introduce el concepto de reconocimiento facial, se describen algunas técnicas, profundizando en la base de datos FERET. Para concluir, se centra en la identificación de fotos de primer plano, que es la dificultad que hay que solucionar en este PFC.

A2.1 Motivación

Una de las necesidades que hay que cubrir con la ficha generada en esta aplicación, es la de proporcionar fotos de primer plano. La BD EMMA tiene contemplada esa posibilidad, pero aún no tenemos fotos etiquetadas como primeros planos. En el apartado 1.3 de la memoria principal veíamos unas gráficas (Fig. 1.1) en la que se detalla que a partir de 2004 cada año se generan entre 150.000 y 250.000 fotografías. Aunque sí tengan etiquetado el nombre del personaje, no existe en el medio periodístico una herramienta que filtre automáticamente fotos de primer plano de un determinado personaje y manualmente sería muy costoso analizar tal cantidad de fotografías. Por tanto es necesaria una herramienta automática con la que analizar las fotos correspondientes al personaje y determinar si son primeros planos o no. Para ello utilizaremos sistemas de reconocimiento facial

A2.2 Introducción

La Real Academia Española define la biometría como el “estudio cuantitativo o estadístico de los fenómenos o procesos biológicos”. Sin embargo, aunque este término nos pueda parecer demasiado vago o impreciso, también se conoce este campo como la utilización de métodos automáticos para el reconocimiento único de humanos, en función de determinados rasgos físicos o de conducta.

Un sistema de reconocimiento facial es una aplicación dirigida por ordenador que identifica automáticamente a una persona en una imagen digital. Esto es posible mediante un análisis de las características faciales del sujeto extraídas de la imagen o de un fotograma clave de una fuente de video, y comparándolas con una base de datos.

La tecnología de reconocimiento facial, al igual que otras técnicas biométricas ha avanzado muchísimo en los últimos años. Hace tiempo, los algoritmos utilizados se basaban en modelos geométricos simples. Sin embargo, las innovaciones computacionales han permitido la creación de una ciencia mucho más sofisticada, basada en lo que se conoce como representaciones matemáticas y procesos de coincidencia.

En los últimos quince años, la tecnología de reconocimiento facial ha dado un salto espectacular, gracias a las novedades presentadas por la industria de este sector, y a

la necesidad de las propias administraciones por adaptar estas técnicas en sus controles policiales y de seguridad.

Lo que conocemos actualmente por biometría facial nació en los años sesenta, con los primeros sistemas que reconocían, gracias a un administrador externo, rasgos como ojos, orejas, nariz o boca, para así tomar distancias de referencia y compararlas con un patrón dado. La automatización del reconocimiento facial no llegaría hasta una década después, cuando se comenzaron a usar características como el grosor de los labios o el color del cabello.

En los años setenta, Goldstein, Harmon, & Lesk, usaron 21 marcadores subjetivos específicos tales como el color del cabello y grosor de labios para automatizar el reconocimiento facial. El problema con estas soluciones previas era que se computaban manualmente. En 1988 Kirby & Sirobich aplicaron análisis de componentes principales, una técnica estándar del álgebra lineal, al problema del reconocimiento facial. Esto fue considerado algo así como un hito al mostrar que eran requeridos menos de 100 valores para cifrar acertadamente la imagen de una cara convenientemente alineada y normalizada. A partir de los noventa surge la biometría facial tal y como la entendemos hoy en día. En 1991 Turk & Pentland utilizan las técnicas *Eigenfaces,* donde el error residual podía ser utilizado para detectar caras en las imágenes; un descubrimiento que permitió sistemas automatizados de reconocimiento facial en tiempo real fidedignos. Si bien la aproximación era un tanto forzada por factores ambientales, creó sin embargo un interés significativo en posteriores desarrollos de estos sistemas. Pero la implementación práctica del reconocimiento facial llegaría en 2001, con la celebración de la Super Bowl de la NFL, donde se archivaron fotografías de los sistemas de vigilancia y se compararon con bases de datos digitales.

Reconocimiento facial y la próxima generación de buscadores

En los últimos años, el desarrollo de nuevas tecnologías informáticas para sistemas de seguridad ha experimentado un gran avance. Entre estos, se destacan los sistemas biométricos para el reconocimiento facial, que se perfilan como los más prometedores.

Ya hace algún tiempo Google hizo un sorprendente anuncio: está desarrollando una técnica que permitirá determinar el sexo de las personas a partir de fotografías utilizando diversos patrones de reconocimiento. Esto supone un primer paso para la búsqueda de información dentro de las imágenes basadas en sistemas inteligentes capaces de determinar qué es lo que las fotos contienen.

El objetivo del buscador era devolver imágenes no en función del texto que les rodea (como se viene haciendo hasta ahora), sino a partir del contenido de la imagen.

Esta fue la siguiente evolución de las herramientas para la indexación de información, compuestas por sistemas expertos que serán entrenados para incrementar la habilidad de determinar, interpretar y organizar el contenido de los documentos (archivos de texto y páginas web, imágenes, vídeos, audio, etc.), superando tecnologías basadas en los tags (como Technorati o del.icio.us) o en los enlaces (Google PageRank).

Google buscó robustecer su servicio y software de gestión para fotografías a través del popular Picasa con la adquisición de Neven Vision en Agostode 2006, una compañía desarrolladora de software para la extracción de información sobre imágenes. El

acuerdo fue anunciado por Adrian Graham, responsable de Picasa, en un blog oficial de Google: "Podría ser tan simple como detectar cuando una foto contiene una persona, o, en un futuro, tan complejo como reconocer gente, sitios y objetos", asegura Graham. La funcionalidad estuvo lista en Septiembre de 2008.

Entendamos que no es fácil buscar a través de las fotos personales, y es ciertamente mucho más duro buscarlo por la web. Para que el Reconocimiento Facial funcione se necesita entrenar a un agente de reconocimiento en primer plano. En primer lugar el agente explorará las fotos o los álbumes seleccionados, y posteriormente enumerará las caras distinguidas en los grupos que representan a una persona. Por supuesto el resultado está lejos de ser perfecto, y necesita inteligencia humana para mejorar los resultados.

Por ejemplo, las fotografías de una persona A se pueden mezclar con una B y así con una tercera o más, en diversos grupos con diversas personas. Entonces necesitaremos nombrar a esos grupos y arrastrar las caras alrededor. Estos grupos formarán una base de datos del reconocimiento facial. Picasa entonces etiqueta las fotos con los tags de los nombres de estos grupos.

Muchos usuarios han estado marcando manualmente las fotos con etiquetas: con este nuevo método para marcar con etiquetas inteligentes a través del reconocimiento facial, se redujo la cantidad de trabajo en un 95%.

Más recientemente, tenemos la noticia de COBOT[18], un exoesqueleto que posee un algoritmo de reconocimiento facial muy mejorado y que estará a la venta en 2016: "En China, un grupo de científicos de la Universidad de Hong Kong ha creado un programa de reconocimiento facial. Los investigadores han desarrollado un algoritmo de sofware a través del que se pueden detectar, analizar y reconocer rostros humanos.

Se puede saber si dos caras que son fotografiadas pertenecen o no a la misma persona, a pesar de los cambios de luz, de maquillaje o del ángulo desde el que se fotografíen.

"Hace un par de meses conseguimos que una máquina superase a los seres humanos en el reconocimiento de rostros. Es algo así como un logro simbólico ya que los investigadores han estado trabajando en esto durante más de 30 años. Es muy emocionante para nosotros" dice el profesor Tang Xiaoou."

En 2014 se anuncia que *Facebook* tiene un sistema de reconocimiento facial más avanzado que el FBI[19]: Tal como lo apunta *The Verge*, durante el verano del hemisferio norte el FBI implementará su sistema de reconocimiento facial llamado *Next Generation Identification*. Esta tecnología se valdrá de una base de datos que contiene 13.6 millones de imágenes de entre 7 y 8 millones de individuos, aunque los números estiman que para el próximo año se tendrán 52 millones de fotos.

El problema del *Next Generation Identification* es la precisión, ya que, de acuerdo a la *Electronic Frontier Foundation*, la efectividad para reconocer a un individuo está lejos de ser óptima. Al someter a reconocimiento una foto se obtendrá una lista de 50 posibilidades y un 85% de probabilidad que aparezca el nombre del sospechoso en la lista. La EFF apunta que las imágenes que se encuentran en las bases de datos de

[18] http://es.euronews.com/2014/07/29/cobot-el-robot-colaborativo-reconocimiento-de-caras-por-ordenador/
[19] http://www.fayerwayer.com/2014/07/facebook-ya-le-gana-al-fbi-en-reconocimiento-facial/

muchas estaciones de policía no cuentan con lo mínimo para ser consideradas. Problemas que van desde resolución baja (0.75 megapixeles es lo recomendado) hasta otras como poca iluminación o interferencia son los causantes de que el reconocimiento facial no sea óptimo.

Comparado con *Facebook* las cosas cambian ya que la red social cuenta con un porcentaje de efectividad del 97.5% para reconocer las caras. El sistema *Deep Face* dice reconocer casi a nivel humano, aunque hay que considerar que las fotografías en *Facebook* se encuentran en mejores condiciones, contrario a aquellas que toman *in fraganti* las cámaras del FBI o aquellas de circuito cerrado colocadas en comercios u otros lugares que terminan por nutrir la base de datos de la agencia federal.

También hay que agregar que la base de datos de *Facebook* es cinco veces más grande que la que tiene el FBI en cuanto a imágenes o que la red social tiene otros medios para potenciar la efectividad, por ejemplo, si hay uno o más amigos de *Facebook* en la foto. Lo cierto es que el FBI está lejos de igualar a *Facebook* en este campo, aunque ya dijo que la lista de resultados que ofrece el Next *Generation Identification* servirá solo como elemento para iniciar una investigación y no para reconocer al culpable.

A2.3 Funcionamiento y técnicas de reconocimiento facial

El objetivo de un sistema de reconocimiento facial es, generalmente, el siguiente: dada una imagen de una cara "desconocida", o imagen de test, encontrar una imagen de la misma cara en un conjunto de imágenes "conocidas", o imágenes de entrenamiento. La gran dificultad añadida es la de conseguir que este proceso se pueda realizar en tiempo real. El sistema identificará las caras presentes en imágenes o videos automáticamente. Puede operar en dos modos:

- Verificación o autentificación de caras: compara una imagen de la cara con otra imagen con la cara de la que queremos saber la identidad. El sistema confirmará o rechazará la identidad de la cara.
- Identificación o reconocimiento de caras: compara la imagen de una cara desconocida con todas las imágenes de caras conocidas que se encuentran en la base de datos para determinar su identidad.

Por su naturaleza amigable, este tipo de sistemas siguen siendo atractivos a pesar de la existencia de otros métodos muy fiables de identificación personal biométricos, como el análisis de huellas dactilares y el reconocimiento del iris.

Funcionamiento

El proceso consta de cuatro módulos principales (Fig. 2.2):

1. Detección de la cara: detecta que hay una cara en la imagen, sin identificarla. Si se trata de un vídeo, también podemos hacer un seguimiento de la cara. Proporciona la localización y la escala a la que encontramos la cara.
2. Alineación de la cara: localiza las componentes de la cara y, mediante transformaciones geométricas, la normaliza respecto propiedades geométricas,

como el tamaño y la pose, y fotométricas, como la iluminación. Para normalizar las imágenes de caras, se pueden seguir diferentes reglas, como la distancia entre las pupilas, la posición de la nariz, o la distancia entre las comisuras de los labios. También se debe definir el tamaño de las imágenes y la gama de colores. Normalmente, para disminuir la carga computacional del sistema, se acostumbra a utilizar imágenes pequeñas en escala de grises. A veces también se realiza una ecualización del histograma.

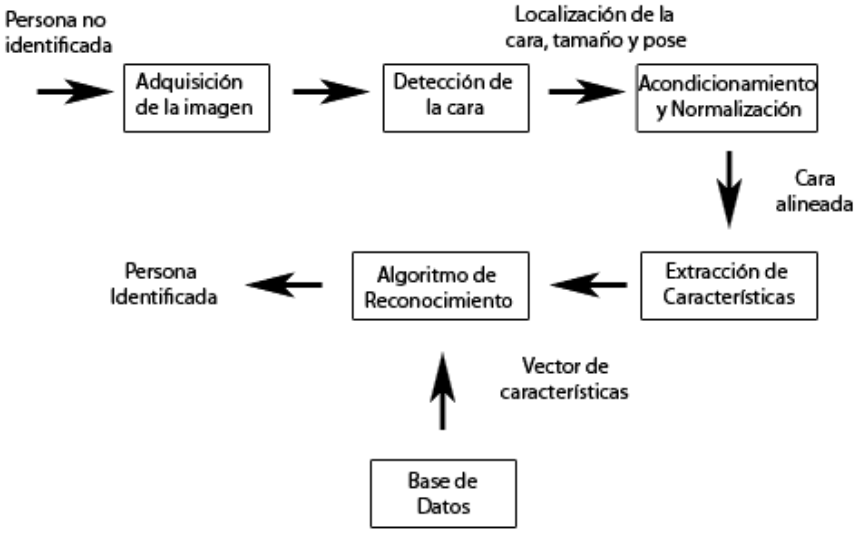

Fig. 2.2 Funcionamiento del reconocimiento facial

3. Extracción de características: proporciona información para distinguir entre las caras de diferentes personas según variaciones geométricas o fotométricas.
4. Reconocimiento: el vector de características extraído se compara con los vectores de características extraídos de las caras de la base de datos. Si encuentra uno con un porcentaje elevado de similitud, nos devuelve la identidad de la cara; si no, nos indica que es una cara desconocida.

Los resultados obtenidos dependen de las características extraídas para representar el patrón de la cara y de los métodos de clasificación utilizados para distinguir los rostros, pero para extraer estas características apropiadamente, hace falta localizar y normalizar la cara adecuadamente.

Técnicas

Entre las técnicas usadas en el reconocimiento facial, podemos destacar las siguientes [BRUPO93]:

- **Sistemas tradicionales**: están basados en la correlación. Van desde la forma más simple, conocido como *template matching,* (donde únicamente se comparan distintos modelos de reconocimiento), o técnicas que utilizan clasificaciones mediante redes neuronales y plantillas deformables.

- **Sistemas locales o geométricos**: en este caso, se analizan vectores

característicos extraídos del perfil del individuo que queremos estudiar, aunque también podemos comprobar los rasgos que pueden observarse de la vista frontal de la cara.

- **Otras técnicas:** los reconocimientos faciales utilizando análisis tridimensionales (mediante sensores especiales) o las técnicas de estudio de textura de la piel, son las novedades más importantes de la biometría facial. En el primer caso se determinan rasgos como la barbilla, el contorno de los ojos o los pómulos. Por otra parte, en el segundo análisis se comprueban detalles como líneas únicas, patrones faciales, manchas o cicatrices.

Dada la gran cantidad de teorías y técnicas aplicables a el reconocimiento facial, son necesarias una clara evaluación y una comparativa para estos algoritmos. Para que sean factibles, se han de utilizar grandes cantidades de imágenes para una evaluación adecuada. También es muy importante que la muestra sea estadísticamente lo más similar posible a las imágenes que surgen en la aplicación que se está considerando. La puntuación ha de realizarse de manera que refleje el coste de los errores de reconocimiento. Recordemos que los resultados dependen mucho de la aplicación que tratamos, así que no podemos extrapolar los resultados para otras aplicaciones.

Existen extensas bases de datos públicas disponibles, así como protocolos de test, para probar las aplicaciones del reconocimiento facial. Uno de ellos es el protocolo FERET, cuyo objetivo es proporcionar un marco de actuación que modele una configuración en tiempo real y reunir una extensa base de datos con imágenes de caras para poder desarrollar algoritmos y evaluarlos. Es el que se utiliza en el FRVT *Face Recognition Vendor Test*.

A2.4 Base de datos FERET

La Base de datos FERET[20] es la norma de de-facto en la evaluación de un Sistema de reconocimiento facial. El programa (FERET) es administrado por la Agencia DARPA *Defense Advanced Research Projects* Agency y NIST *National Institute of Standards and Technology*. Consiste en una base de datos de imágenes faciales que se recogió entre diciembre de 1993 y agostode 1996. En 2003 publicó una versión de alta resolución, 24 bits de color, de estas imágenes. El conjunto de datos incluye 2413 imágenes faciales, representando a 856 personas.

El programa fue establecido para crear una gran base de datos de imágenes faciales que se obtuvo de forma independiente para poder evaluar los algoritmos de reconocimiento facial.

En la publicación de 1999 “The FERET Evaluation Methodology for Face-Recognition Algorithms” [FERET99] se nos explican tanto los test para el reconocimiento facial como la base de datos FERET creada para contrastar las imágenes.

[20] http://www.frvt.org/FERET/default.htm

A2.5 Identificación de una foto de primer plano

En el caso del proyecto que nos ocupa, nos interesa, teniendo fotografías ya etiquetadas con el nombre del personaje del que estamos buscando información, identificar cuáles son un primer plano. En un futuro, esta etiqueta será parte de la información almacenada en la base de datos del medio informativo, pero de momento no disponemos de esas etiquetas. Por lo tanto, tenemos que encontrar un algoritmo que nos permita discriminar este tipo de fotos.

Una fotografía de primer plano deja ver el rostro y los hombros. Implica cierto grado de intimidad y confidencialidad, así que con su uso podremos transmitir emociones más intensas que con los demás.

Otra variante es la fotografía de primerísimo primer plano: abarca un rostro desde el mentón hasta la parte de arriba de la cabeza. Transmite incluso más intimidad y confidencialidad que el primer plano.

Para el problema de detectar fotografías de primer plano, podemos utilizar los primeros pasos de la técnica utilizada en las cámaras digitales, ya que el reconocimiento facial ya es una característica de la mayoría de ellas.

Para el proceso de reconocimiento facial que nos interesa, utilizaríamos dos pasos: El primero, que también es el que utiliza más tiempo de procesamiento, es la detección de la cara. Un cuadro de detección de al menos 20 x 20 píxeles barre todo el espacio. La detección se basa en la edad, nacionalidad, así como la orientación y la dirección de la cara. No hay límite en cuanto al número de caras que se pueden detectar, sin embargo, esto también significa más tiempo de procesamiento.

El segundo paso sería encontrar ciertas partes de la cara, en la cara de cada individuo, generalmente son cuatro: el ojo izquierdo y derecho, la nariz y la boca.

Ya que en nuestro medio de comunicación las fotos ya están etiquetadas con el nombre del personaje al que pertenecen, y sólo necesitamos determinar si la fotografía es de un primer plano, utilizaríamos estos dos pasos para identificar que sólo hay una cara en la foto, y valorar, según el tamaño de la cara respecto al tamaño de la foto, si ésta es el primer plano que buscamos o no.

A3. WEB SEMÁNTICA

Entre las últimas tendencias que pueden repercutir en el futuro de la web a medio plazo, a finales de los 90 surge la visión de lo que se ha dado en llamarla Web Semántica [Berners-Lee 2001]. Se trata de una corriente, promovida por el propio inventor de la web y presidente del consorcio W3C12, cuyo último fin es lograr que las máquinas puedan entender, y por tanto utilizar, lo que la web contiene. Esta nueva web estaría poblada por agentes o representantes software capaces de navegar y realizar operaciones por nosotros para ahorrarnos trabajo y optimizar los resultados.

A3.1 Motivación

Para conseguir esta meta [CAS03], la Web Semántica propone describir los recursos de la web con representaciones procesables (es decir, entendibles) no sólo por personas, sino por programas que puedan asistir, representar, o reemplazar a las personas en tareas rutinarias o inabarcables para un humano. Las tecnologías de la Web Semántica buscan desarrollar una web más cohesionada, donde sea aún más fácil localizar, compartir e integrar información y servicios, para sacar un partido todavía mayor de los recursos disponibles en la Web.

Uno de los principales obstáculos que nos encontramos para realizar las fichas biográficas es el reconocimiento de patrones para localizar determinados datos que se encuentran embebidos en textos o se encuentran en Internet de forma desestructurada; así como la búsqueda de información sobre el personaje en un repositorio semántico, como es DBpedia. Por lo tanto, es importante para nuestro proyecto conocer conceptos relacionados con la web semántica ya que nos aportan soluciones para el desarrollo de este PFC.

El artículo “Evolución y uso de los lenguajes controlados en documentación informativa” [CACU07] nos explica que la documentación periodística adolece de instrumentos adecuados de clasificación e indexación de la información, a excepción del *Subject Reference System del Internacional Press Telecommunications Council* (IPTC), todavía en estado incipiente de desarrollo. También nos hace una revisión de las contribuciones más relevantes sobre la clasificación e indexación de noticias en los medios de comunicación, tanto españoles como internacionales, estudia la elaboración y uso de vocabularios controlados para el tratamiento de la información de actualidad y sobre todo de tesauros especializados (como es el caso del tesauro utilizado en el medio de comunicación en el que desarrollamos este proyecto).
Por último, se destacan algunas características específicas de la documentación periodística que condicionan la utilización de tesauros para la indexación y recuperación de información de actualidad.

Este artículo es interesante porque da una visión global del estado del uso de tesauros y otro tipo de vocabularios controlados en el almacenamiento del medio periodístico. Esta visión global ayuda a contextualizar el método utilizado en el Heraldo de Aragón dentro de la situación general.

A3.2 Introducción

La W3C (World Wide Web Consortium) nos define la Web Semántica como “una Web extendida, dotada de mayor significado en la que cualquier usuario en Internet podrá encontrar respuestas a sus preguntas de forma más rápida y sencilla gracias a una información mejor definida. Al dotar a la Web de más significado y, por lo tanto, de más semántica, se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla. Esta Web extendida y basada en el significado, se apoya en lenguajes universales que resuelven los problemas ocasionados por una Web carente de semántica en la que, en ocasiones, el acceso a la información se convierte en una tarea difícil y frustrante.”

El precursor de la idea, Tim Berners-Lee, intentó desde el principio incluir información semántica en su creación, la World Wide Web, pero por diferentes causas no fue posible. Por ese motivo introdujo el concepto de semántica con la intención de recuperar dicha omisión.

A3.3 Conceptos de Web Semántica

Analizando la convergencia que se está produciendo en el campo de las ontologías entre ingeniería del conocimiento y organización del conocimiento en el marco de la Web Semántica, y estudiando el desarrollo de la investigación sobre ontologías en las ciencias de la documentación y en el conjunto de las disciplinas que se interesan por los problemas ontológicos, encontramos el artículo “Ontologías y organización del conocimiento: retos y oportunidades para el profesional de la información” [GARMAR07]. Aquí se contextualiza el actual frente de investigación en el campo de las ontologías en el marco del desarrollo de Internet y especialmente de la Web Semántica, y se analizan las implicaciones de futuro para el profesional de la información, viendo que para ellos, la integración en el campo de la Web Semántica, el clarificar su posición en él, y asegurar una formación adecuada en los nuevos estándares y tecnologías, es básico. Otro artículo relevante es “A translation approach to portable ontology specifications” [GRU93] donde se explica un método para definir ontologías.

El objetivo de las ontologías es constituir un almacén de información semántica donde sea posible consultar el significado de un término a través de los mecanismos proposicionales propuestos ya por Aristóteles en su teoría de la definición. Los significados de los términos son resueltos por los expertos en ontologías a partir de las relaciones entre los mismos.

Hasta ahora la WWW ha facilitado enormemente el proceso de compartir información entre personas gracias a sus eficaces estándares de comunicación
-HTTP– y de normalización de los documentos –HTML–. Sin embargo, no permite la recuperación y procesamiento de la información a nivel de dato y combinaciones de ellos (información), sino tan sólo del documento, lo que es imprescindible para los procesos de automatización que soportan los diferentes tipos de lenguajes de programación.

En cambio, la Web Semántica trata de dos cosas. Se ocupa de que existan formatos comunes de intercambio de datos, mientras que en la web original sólo teníamos intercambio de documentos. Además tiene que ver con la elaboración de un lenguaje

para codificar cómo los datos se relacionan con los objetos del mundo real. Eso permitiría a una persona o a una máquina comenzar a trabajar en una base de datos, y luego moverse a través de otras que no están conectadas por cables, sino por tratar del mismo asunto (web oficial W3C[21], 1994-2004).

A3.3.1 Definición

La Web Semántica es una ampliación de la Web, por medio de la que se intenta realizar un filtrado de manera automática pero precisa de la información. Es necesario hacer que la información que anida en la web sea entendible por las propias máquinas. En concreto se atiende a su contenido, independientemente de la estructura sintáctica. O lo que es lo mismo, se atiende a diferentes ámbitos, se tiene en cuenta el conjunto de lenguajes, a la vez que los procedimientos para poder añadir esa semántica a la información para que, de esta manera, sea entendible por los agentes encargados de procesarla. Además, se tiene en cuenta el desarrollo y la construcción de los agentes encargados de procesar esa información y de filtrar adecuadamente cuál de todas ellas es la útil para los usuarios o para los agentes que tienen que realizar una función concreta. Con todo ello, los agentes deben recuperar y manipular la información pertinente, lo que requiere una integración sin fracturar la web, pero sin dejar de aprovechar totalmente las infraestructuras que existen. En concreto, a través de esta modalidad de Web Semántica se pueden obtener soluciones a problemas habituales en la búsqueda de información gracias a la utilización de una infraestructura o proceso común, mediante la cual, es posible compartir, procesar y transferir información de forma sencilla.

En la actualidad, la World Wide Web está basada principalmente en documentos escritos en HTML, un lenguaje de marcado que sirve principalmente para crear hipertexto en Internet. El lenguaje HTML es válido para adecuar el aspecto visual de un documento e incluir objetos multimedia en el texto (imágenes, esquemas de diálogo, etc.). Pero ofrece pocas posibilidades para categorizar los elementos que configuran el texto más allá de las típicas funciones estructurales, como sucede con otros lenguajes de maquetación (tipo LaTeX).

HTML permite mediante una herramienta de visualización (como un navegador o un agente de usuario) mostrar por ejemplo un catálogo de objetos en venta. El código HTML de este catálogo puede explicitar aspectos como "el título del documento es Ferretería Acme"; pero no hay forma de precisar dentro del código HTML si el producto M270660 es una "batería Acme", con un "precio de venta al público" de 200 €, o si es otro tipo de producto de consumo (es decir, es una batería eléctrica y no un instrumento musical, o un puchero). Lo único que HTML permite es alinear el precio en la misma fila que el nombre del producto. No hay forma de indicar "esto es un catálogo", "batería Acme" es una batería eléctrica, o "200 €" es el precio. Tampoco hay forma de relacionar ambos datos para describir un elemento específico en oposición a otros similares en el mismo catálogo.

La Web Semántica se ocuparía de resolver estas deficiencias. Para ello dispone de tecnologías de descripción de los contenidos, como RDF y OWL, además de XML, el lenguaje de marcado diseñado para describir los datos. Estas tecnologías se combinan para aportar descripciones explícitas de los recursos de la Web (ya sean estos catálogos, formularios, mapas u otro tipo de objeto documental). De esta forma el contenido queda desvelado, como los datos de una base de datos accesibles por

[21] http://www.w3.org/2001/sw/

Web, o las etiquetas inmersas en el documento (normalmente en XHTML, o directamente en XML, y las instrucciones de visualización definidas en una hoja de estilos aparte). Esas etiquetas permiten que los gestores de contenidos interpreten los documentos y realicen procesos inteligentes de captura y tratamiento de información.

A3.3.2 Avances

Actualmente, existen nichos piloto que han comenzado con la transformación hacia la Web Semántica:

- Sistemas de Datos Abiertos gubernamentales en varios países, se encuentran en formato RDF.
- Datos Abiertos en la Biblioteca Nacional de Francia > data.bnf.fr
- Intranets de conocimiento de empresas multinacionales.
- Incorporación de metadatos en sistemas de comercio electrónico.
- Resultados semánticos en el motor de búsquedas Google[22]. Para proporcionar datos más acordes a las consultas realizadas a través de su buscador, Google utilizará una gran base de datos integrada por la información recopilada durante los dos últimos años de las búsquedas de los usuarios. Estos datos incluyen personas, lugares y cosas, a los que se añaden los de las bases de datos de la compañía Metaweb Technologies que Google compró en 2010. La adopción de la tecnología semántica se producirá gradualmente, lo que provocará variaciones en el algoritmo utilizado por la compañía para asignar el PageRank a las páginas web. Una vez desplegadas, las búsquedas semánticas permitirán obtener información directa, además de enlaces a las páginas web más relevantes para el término buscado. Si la búsqueda se refiere, por ejemplo, a Islandia, se ofrecerán datos sobre el país, además de enlaces a las páginas web en las que se haga referencia a él.
- DBpedia es un proyecto para la extracción de datos de Wikipedia para proponer un repositorio semántico.

A3.3.3 Componentes y arquitectura tecnológica de la Web Semántica

Los principales componentes de la Web Semántica son los metalenguajes y los estándares de representación XML, XML Schema, RDF, RDF Schema y OWL, así como el lenguaje SPARQL para la consulta de datos RDF [4 (Ver Fig. 2.3)]. En la página de *OWL Web Ontology Language Overview*[23] se describe la función y relación de cada uno de estos componentes de la Web Semántica:

- XML aporta la sintaxis superficial para los documentos estructurados, pero sin dotarles de ninguna restricción sobre el significado.
- XML Schema es un lenguaje para definir la estructura de los documentos XML.
- RDF es un modelo de datos para los recursos y las relaciones que se puedan establecer entre ellos. Aporta una semántica básica para este modelo de datos que puede representarse mediante XML.
- RDF Schema es un vocabulario para describir las propiedades y las clases de los recursos RDF, con una semántica para establecer jerarquías de generalización entre dichas propiedades y clases.
- OWL es un lenguaje para definir ontologías mediante la descripción detallada de propiedades y clases: tales como relaciones entre clases (p.ej. disyunción), cardinalidad (por ejemplo "únicamente uno"), igualdad, tipologías de

22 http://www.pcactual.com/articulo/actualidad/noticias/10604/google_mejorara_los_resultados_con_tecnologia_semantica.html#sthash.Iq8RdG5v.dpuf

23 http://www.w3.org/TR/owl-features/

propiedades más complejas, caracterización de propiedades (por ejemplo simetría) o clases enumeradas.

- SPARQL es un lenguaje de consulta de conjuntos de datos RDF. Además en dicha especificación también se incluye un formato XML que detalla el modo en el que se estructuran los resultados obtenidos.

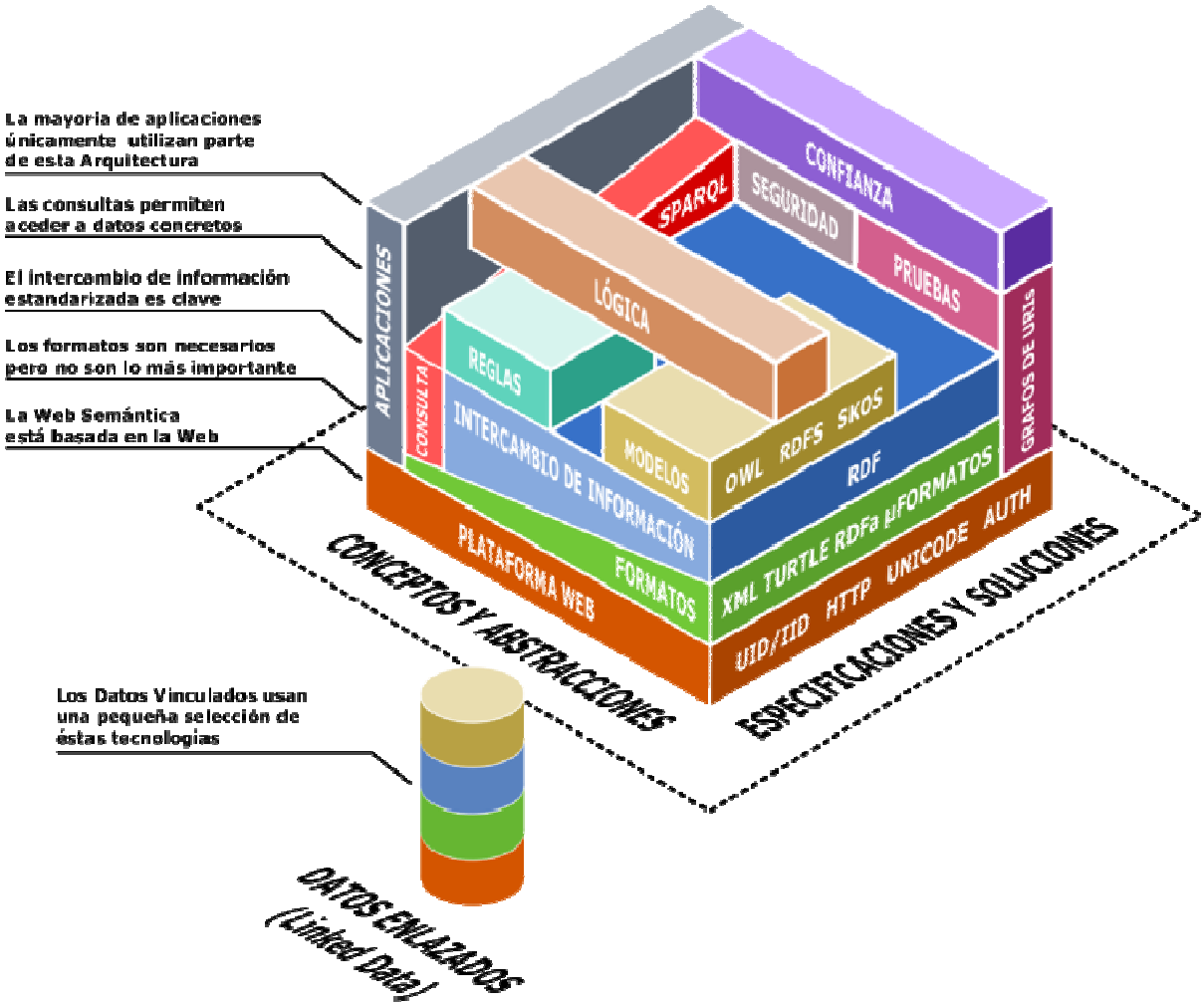

Fig. 2.3 Componentes de la Web Semántica. © CC BY-SA 3.0

Los proveedores primarios de esta tecnología son las URIs (*Uniform Resource Identifier – Identificador de recursos uniforme, que identifica de forma unívoca)* que identifican los recursos junto con XML y los *namespaces*. Si a esto se añade un poco de lógica mediante RDF, u otras tecnologías como los mapas temáticos y algo de razonamiento basado en técnicas de inteligencia artificial, Internet podría estar cerca de alcanzar las aspiraciones iniciales de su inventor, Tim Berners-Lee.

A3.4 Linked Data

La propuesta de datos enlazados o *linked data* surge dentro de marco general de la Web Semántica. El término "*linked data (*datos enlazados)" hace referencia al método con el que se pueden mostrar, intercambiar y conectar datos a través de URIs referenciables en la Web.

En el artículo “A Database Perspective on Consuming Linked Data on the Web” [HARLANG10] se nos presenta el concepto de “Linked Data (LD)” y cómo utilizarla en la Web aplicando diferentes aproximaciones con consultas cercanas al SQL.

Los principios de “Linked Data” son los siguientes:

- Utilizan las URIs como nombres para las entidades
- Utilizan las URIs de manera que la gente pueda buscar esos nombres
- Cuando alguien busca una URI, proporciona información útil, utilizando estándares (RDF, SPARQL)
- Incluyen links a otras URIs (dentro de su información) de manera que se pueda descubrir otras entidades.

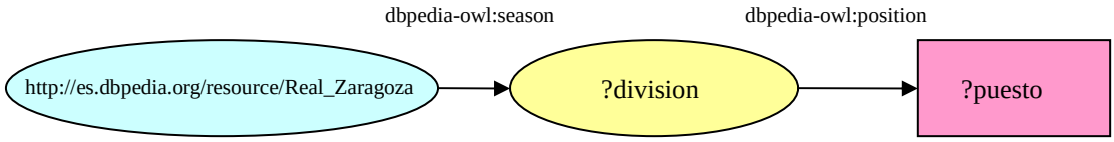

Figura 2.4: Gráfico que representa un patrón básico con dos variables en la consulta: ?division y ?puesto

Este artículo nos aporta una visión introductoria al SPARQL como estándar para consultas a la Web Semántica.

Algunos ejemplos del uso de *linked data* en la web son: Freebase[24], OpenCyc[25] o DBpedia.

A3.5 DBpedia

DBpedia es un proyecto para la extracción de datos de Wikipedia para proponer una versión Web Semántica. Este proyecto está realizado por la Universidad de Leipzig, Universidad Libre de Berlín y la compañía OpenLink Software.

DBpedia está interconectada con GeoNames, Musicbrainz, CIA World Factbook, Proyecto Gutenberg y Eurostat entre otros.

[24] https://www.freebase.com/
[25] http://www.cyc.com/platform/opencyc

En la base de datos se describen 3.640.000 entidades, entre ellas al menos 416.000 personas, 526.000 lugares, 106.000 álbumes de música y 60.000 películas y contiene 2,724,000 enlaces a imágenes, 6,300,000 enlaces a páginas externas, 6,200,000 enlaces a datasets externos y 740,000 categorías Wikipedia.

El contenido de la base de datos está disponible bajo licencia CC-BY-SA 3.0 y GFDL (ya que el contenido se basa en la Wikipedia).

En mayo de 2012 se lanzó el sitio web de DBpedia para el idioma español[26].

La información se almacena con *Resource Description Framework* (RDF)[27], y podemos hacer consultas a la base de datos a través de SPARQL.

Olaf Hartig justifica la utilización de SPARQL [HAR11] como lenguaje de consulta para Linked Data en la Web Semántica. En los últimos 20 años, los links se habían hecho mediante hipertexto. En este artículo investigan formalmente la utilidad de SPARQL que se ha demostrado en la práctica como uno de los lenguajes de consulta más importantes.

En el artículo "Executing SPARQL Queries over the Web of Linked Data**"** [HARBIZFREY09] también nos presentan cómo se ejecutan las consultas SPARQL sobre la web de *Linked Data*. La idea principal es descubrir los datos que pueden ser relevantes para responder a una consulta durante la propia ejecución de la consulta. Este descubrimiento se hace siguiendo links RDF entre fuentes de datos basadas en URLs que se incluyen en la consulta y en resultados parciales. Las URLs se resuelven mediante el protocolo HTTP en datos RDF que se añaden al conjunto de datos consultados.

El motor de extracción de datos se realiza con Scala, un software libre publicado bajo el GNU General Public License. Su código fuente se distribuye: se alberga en Sourceforge y está disponible a través de Subversion.

El proyecto DBpedia ha generado durante mucho tiempo información semántica a partir de la wikipedia inglesa. Desde junio de 2011 el proceso de generación de información extrae infomación de Wikipedia en 15 de sus versiones (idiomas). Uno de ellos es el español. El comité de internacionalización de DBpedia ha asignado un sitio web y un SPARQL EndPoint para cada uno de estos idiomas. Un SPARQL Endpoint es el lugar donde puedes consultar la información almacenada. Por ejemplo, el SPARQL endpoint de es.dbpedia.org está aquí: http://es.dbpedia.org/sparql. En el SPARQL endpoint están disponible los triples más relevantes (~70 millones).

La DBpedia en español depende principalmente de investigadores de la UAM (Mariano Rico) y la UPM (Óscar Corcho), todos ellos pertenecientes a la Red Temática Española de *Linked Data*, así como de particulares que dedican su tiempo y su esfuerzo a esta iniciativa.

[26] http://es.dbpedia.org
[27] http://www.w3.org/TR/RDF

A3.6 Bases de datos semánticas

En 1970, el Modelo Relacional revolucionó el campo de las Bases de Datos, debido al logro de la separación de la representación lógica del dato de la implementación física, lo que produjo en adelante el desarrollo de lenguajes de consultas. El artículo "**Un breve panorama sobre las Bases de Datos Semánticas"** [SHI03] nos explica que la historia del modelamiento semántico, también data hacia esas fechas. Los modelos semánticos fueron introducidos como herramientas de diseño de esquemas. El motivo principal de su uso radicaba en la exactitud del modelo de datos. Como muchos autores aseveran, el modelado semántico no es más que una representación del mundo real.
Estas propiedades no las tenían las aplicaciones de bases de datos típicas.
El primer modelo semántico publicado apareció en 1974. El área maduró con el desarrollo de muchos modelos importantes —como Semantic Data Model , General Semantic Data Model, Iris Data Model, Sem-Object Data Model, IFO—. Después se ha tratado de estandarizar todo, con la llegada del estándar SQL3 en el año 1999. En décadas anteriores entre 1970 y 1980 se planteaba una división entre el modelo conceptual —modelado semántico— y el modelo lógico —modelado relacional—. Para 1990 Naphtali Rishe [RISYUATH00] planteaba un modelo que sugería colocar el nombre de Semantic-SQL, que pudiera ser un estándar en lenguajes de consultas semánticas y planteaba una arquitectura del sistema donde el modelo relacional actuaba de forma independiente al modelo semántico. Este último actuaba de manera directa con el usuario y traducía la consulta a un modelo lógico entendible por el Sistema de Base de Datos Relacional.

El ejemplo real que conecta este proyecto con la evolución de las Bases de Datos semánticas, es OpenLink Virtuoso[28]. Éste es el motor de base de datos que utiliza DBpedia en su página web[29] para realizar consultas y que originalmente extendió su implementación en SQL3 con sintaxis para integrar SPARQL.

A3.7 Conclusiones

Valorando la importancia de este capítulo en el desarrollo del PFC quiero destacar que este estudio del arte ha sido necesario en primer lugar, para una formación personal sobre los temas que había que aplicar en el desarrollo de esta herramienta. Estas lecturas han sido muy importantes para actualizar lo que conocía sobre estos temas.

Además han aportado las ideas para las soluciones desarrolladas, ya que combinando técnicas ya existentes y adaptándolas a las necesidades concretas que surgen en este caso se han podido desarrollar los algoritmos que resuelven las mayores dificultades.

Viendo los resultados de la aplicación de los algoritmos estudiados, y de las posibilidades prácticas de aplicarlos, se escogen dos soluciones: una, optar por la implementación utilizando una desambiguación basada en un catálogo previo de *Named Entities* y el tesauro del archivo documental. Y en segundo lugar, apoyarse en la búsqueda dentro del repositorio semántico que es DBpedia, como algo novedoso y que puede aportar datos que faciliten la desambiguación.

[28] http://virtuoso.openlinksw.com/
[29] http://es.wikipedia.org/wiki/SPARQL

También hay que tener en cuenta que en este caso, no se necesitan resolver todas las correferencias, sino sólo las del personaje que buscamos. Así que habrá que crear un algoritmo más sencillo que los estudiados en estos capítulos.

ANEXO B: GENERADOR AUTOMÁTICO DE FICHAS DE PERSONAJES

La labor de los documentalistas, además del correcto almacenamiento de la información, consiste en proporcionar a los periodistas los datos que necesiten para elaborar los nuevos artículos. Esta información puede ser un texto, o una imagen almacenadas en la base de datos documental a través de EMMA.

La herramienta que se presenta aquí trata de facilitar el trabajo del documentalista, ayudando en el caso concreto de que éste busque información sobre un personaje. El generador de fichas busca información (incluyendo imágenes de distintos tipos) relevante sobre este personaje que el periódico tenga almacenada, filtrando y desechando la irrelevante o errónea, y ordenándola cronológicamente. Además completa el resultado proporcionando datos biográficos y enlaces de Internet y DBpedia.

B1. CONTEXTO

El generador de fichas biográficas nace como un prototipo que se desarrolla para funcionar en los equipos de periodistas y documentalistas de manera independiente, y termina formando parte de las herramientas que ofrece la plataforma EMMA.

La plataforma EMMA (*Entorno Multimedia de Archivo*) es un entorno de trabajo pensado para almacenamiento y gestión de un gran volumen de información digital editorial, dentro del contexto empresarial de un medio de comunicación escrito. Ha sido desarrollada en la empresa Ibercentro Media Consulting & Services S.L. y la distribuye Hiberus[30], una conocida empresa aragonesa de Tecnologías de Información (fig. 3.1)

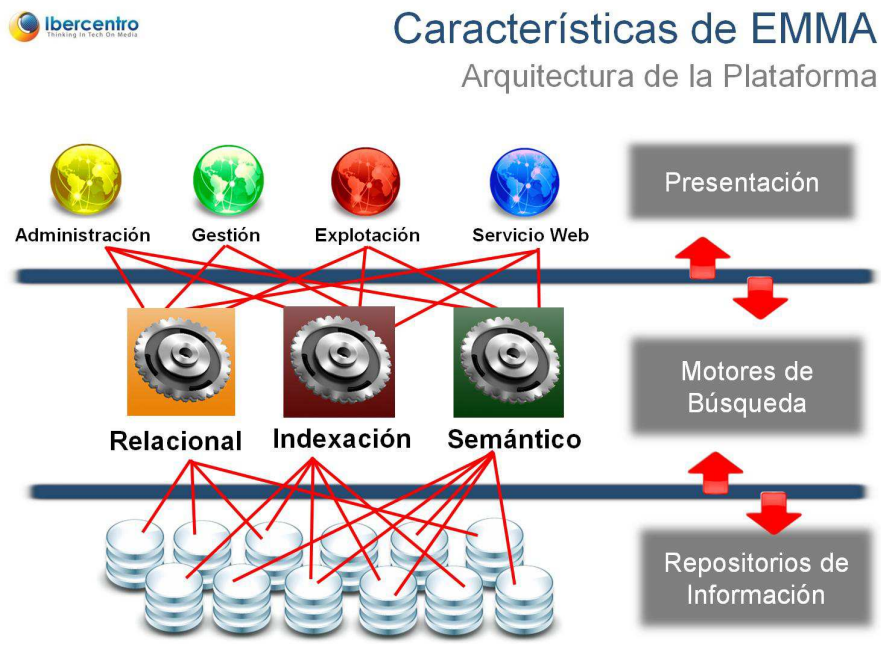

Figura 3.1 Esquema de arquitectura de la plataforma EMMA

[30] *Hiberus: http://www.hiberus.com*

Las herramientas que permiten la búsqueda de información almacenada en el medio periodístico se encuentran dentro de la plataforma EMMA.

El personal de documentación, encargado de proporcionar toda la información que les es requerida por parte de los periodistas, pierde una gran cantidad de tiempo recopilando información del archivo documental y de Internet. Este proyecto será una herramienta más integrada en la plataforma EMMA de cara a ser accesible directamente por los periodistas y documentalistas.

B2. ANÁLISIS

A continuación en este capítulo, se pasarán a describir las necesidades que se han presentado para el desarrollo del generador automático de fichas biográficas y explicar lo que debería hacer el software para satisfacer dicha necesidad.

B2.1 Requisitos previos

La aplicación desarrollada en este PFC debe ser una herramienta software que permita generar de forma automática fichas informativas estructuradas, centrándose en el caso concreto de los personajes que aparecen en las noticias. Partiremos de los textos y fotos pertenecientes a una base de datos documental del medio de comunicación periodístico, pero también utilizaremos otras fuentes de Internet, incluyendo un repositorio semántico, como es DBpedia.

Este proyecto se desarrolla en Ibercentro Media Consulting & Services S.L., que es una empresa que presta servicio informático al Grupo Heraldo y Diario de Navarra.

El objetivo principal es desarrollar un prototipo funcional que forme parte de la plataforma de documentación, enmarcado en concreto entre los procedimientos de búsqueda de información de que actualmente funcionan sobre la plataforma documental de contenidos de Heraldo de Aragón y Diario de Navarra.

La herramienta, en una primera fase como prototipo, será sólo accesible por el personal de este departamento, y más adelante se integrará en la plataforma EMMA que da servicio al resto de personal de edición y archivo.

El generador de fichas extraerá la información de tres fuentes: por un lado, de la Base de Datos EMMA, donde están archivados todos los artículos publicados en los últimos años, y las referencias a los archivos PDF y de imagen que pueden ser necesarios. En segundo lugar, deberá tener acceso a Internet para buscar información complementaria (en esta herramienta se buscarán sobre el personaje páginas web personales, libros, vídeos, imágenes, blogs y perfiles en redes sociales), y finalmente, de la misma fuente aunque no del mismo modo, poder extraer información del proyecto de repositorio semántico DBpedia.

B2.2 Análisis de requisitos

Aquí se recogen las necesidades mostradas desde el periódico y asumidas para el desarrollo de este PFC.

Objetivos funcionales

Para cumplir sus objetivos, el generador de fichas biográficas deberá realizar las siguientes funciones:

- Realizar correctamente la conexión a todas sus fuentes de datos (EMMA, Internet).

- Capturar los datos de entrada. En este caso, habrá una interfaz de entrada para el usuario, que transformará esos datos en un archivo XML. En ese archivo se incluyen los siguientes datos:

    - Nombre del personaje a buscar
    - Extensión del archivo resultante
    - Elección entre noticias y fotos embebidas en el texto, o disponibilidad de enlaces a ellas.

- Proporcionar un primer filtrado de la información disponible para la toma posterior de decisiones.

- Desambiguar el nombre del personaje en caso de coincidencias y obtener el nombre correcto y completo para realizar búsquedas.

- Obtener la información requerida de la base de datos de archivo. Pueden ser fotografías, noticias, páginas o infografías.

- Obtener la información disponible en Internet. Pueden ser enlaces, imágenes, vídeos, libros, blogs, referencias en redes sociales, o páginas web personales.

- Obtener la información disponible en DBpedia.

- Generar una salida. En este caso, los resultados se ofrecen en un archivo XML con los datos estructurados que finalmente se transformará en un HTML con una extensión ajustada a los requisitos de entrada.

- Realizar un control de la concurrencia: varios usuarios pueden requerir la elaboración de una ficha de personaje a la vez.

Antes de la versión definitiva y para crear un prototipo, se realiza además un interfaz de entrada para la captura de datos directa desde el usuario, y un visualizador que muestra en pantalla los resultados obtenidos.

Requisitos no funcionales

Los requisitos técnicos son los siguientes para el equipo de pruebas donde se ejecuta el sistema (equipo que ya existe en el medio periodístico donde se ha desarrollado el PFC):

Sistema operativo	Microsoft Windows XP Professional
Versión	5.1.2600 Service Pack 3 Compilación 2600
Fabricante sistema operativo	Microsoft Corporation
Tipo de sistema	Equipo basado en X86
Procesador	x86 Family 15 Model 4 Stepping 3 GenuineIntel ~3600 MHz
BIOS Versión/Fecha	American Megatrends Inc. 0411, 28/10/2008
Memoria física total	1.024,00 MB
Memoria virtual total	2,00 GB

- Plataformas soportadas: Windows.
- Paralelismo: no lo hay por el momento, toda la carga de trabajo del procesamiento de un texto la realizará solamente un equipo.
- Consideraciones sobre usuarios y seguridad: La aplicación sólo considerará permisos para acceder a la configuración de la aplicación, y éstos serán otorgados únicamente al personal técnico informático.

Requisitos empresariales

Los objetivos a nivel de organización son básicamente dos:

-Reducir el tiempo de búsqueda de información sobre un personaje concreto,

-Facilitar la labor del personal de documentación y periodistas.

B2.3 Descripción del sistema

A continuación se muestra un diagrama de despliegue del sistema (Fig 2.2). El sistema está compuesto principalmente por tres librerías que se comunican entre sí a través de archivos XML.

En primer lugar, el usuario periodista o documentalista, se conectará a través de una aplicación cliente, a través de la plataforma web EMMA (tal como vienen haciendo ya para otros servicios de EMMA) al servidor principal de EMMA, que será el encargado de llamar a la primera librería (BGRPHY_LIB), encargada de realizar las búsquedas en Internet y en el archivo documental (almacenado en la base de datos SQL Server). Se apoyará en una segunda librería (BGRPHY_DBPEDIA), que realiza las búsquedas en DBpedia. Finalmente, la tercera librería (BGRPHY_IO) será la que de formato a los resultados para presentarlos al usuario. Estas tres librerías se desarrollan en este PFC.

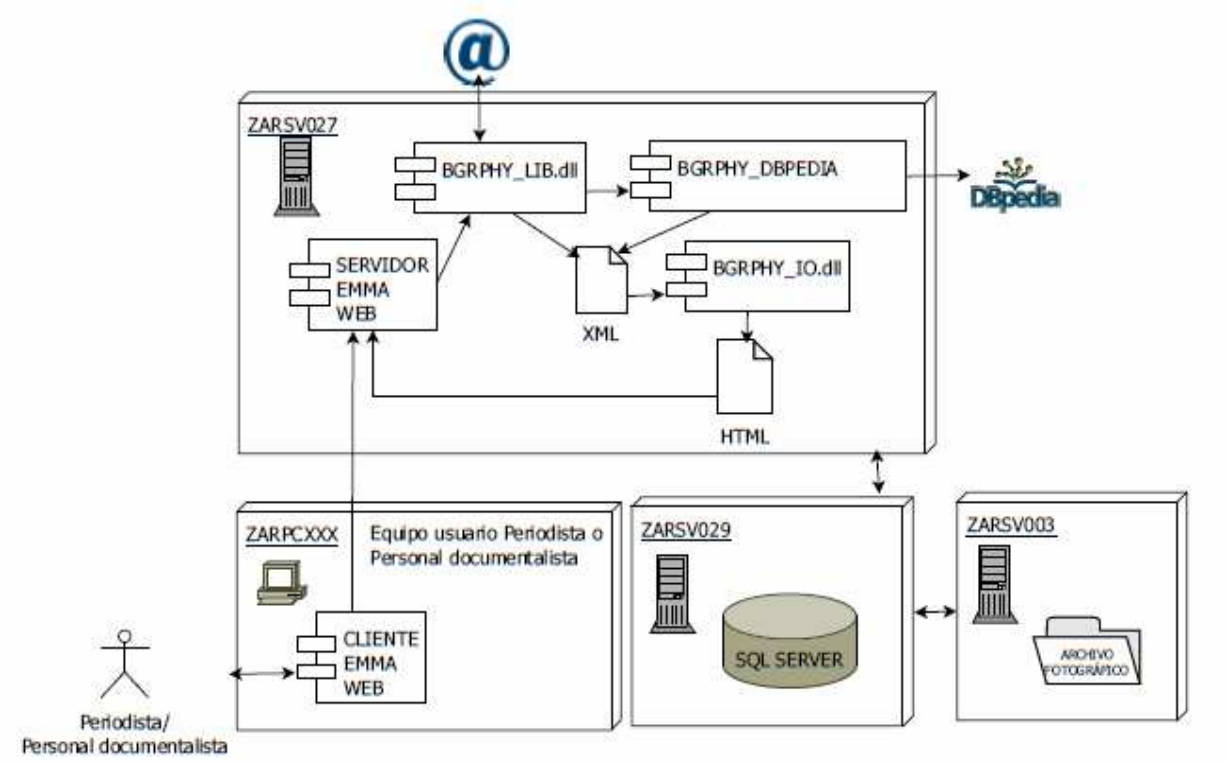

Figura 2.2: Diagrama de despliegue

A continuación aparece el diagrama de actividades del sistema (Fig. 2.3), que muestra el orden en que se van produciendo las distintas operaciones. En él se ve que el sistema realiza en primer lugar una búsqueda previa, para analizar la presencia del personaje en el tesauro del archivo documental y en DBpedia. Si la presencia no es relevante, se descarta la búsqueda del personaje. Si es relevante, y hay problemas de ambigüedad (coincidencia de dos o más entidades con el mismo nombre), lo resuelve.

Con un nombre de personaje definido tras esta primera etapa, se comienza la búsqueda de toda la información que formará parte de la ficha biográfica, en primer lugar en el archivo documental (noticias, fotografías), a continuación en Internet (datos biográficos en Wikipedia, enlaces a blogs, webs, libros, otras noticias, vídeos, redes sociales). Finalmente completa los datos restantes consultando DBpedia en español.

Con toda esta información se genera un archivo XML que finalmente se transformará en uno HTML para presentación al usuario. Estos dos pasos son necesarios ya que el fichero XML nos da un resultado que es independiente de si se va a mostrar en un solo equipo mediante un visualizador para los resultados, o si se integra en una plataforma web como es EMMA.

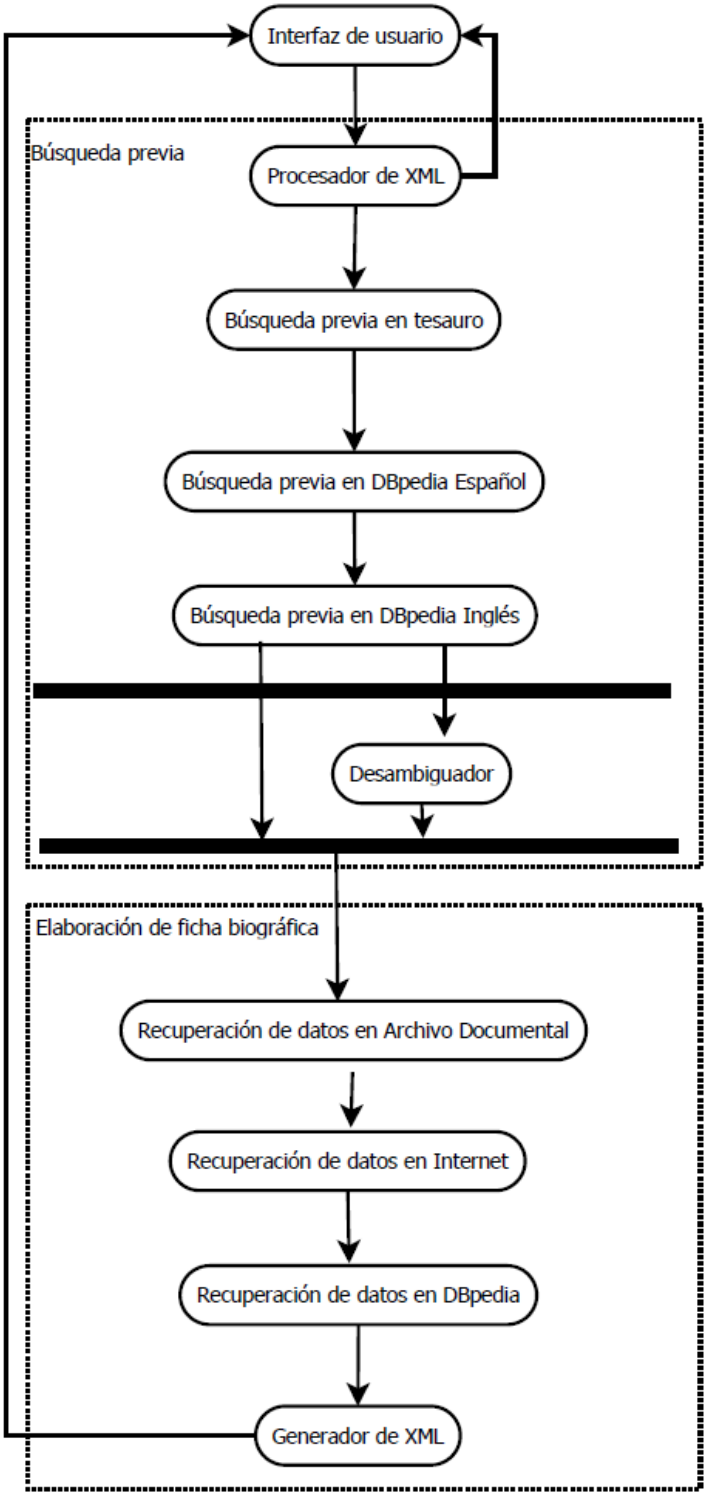

Figura 2.3: Diagrama de actividades

B2.4 Diagrama de clases

En este apartado se describen todas las clases que se han creado para este PFC. Se han repartido en tres diagramas distintos, porque describen conjuntos de clases que no tienen relación entre sí y que se usan en módulos diferentes de la herramienta.

En el primer diagrama (Fig. 3.5) se describe el objeto datos_personaje, que contiene su nombre, apellidos y alias (proporcionados por el usuario en un principio, resultado de la desambiguación en otros casos); otros atributos corresponden a datos necesarios sobre cada personaje para proceder a su desambiguación.

La subclase datos_iniciales hereda los atributos de datos_personaje, añadiendo los atributos que contienen el nombre completo y un procedimiento (datos_incompletos) que nos permite saber si el usuario ha completado la información requerida para que comience el proceso, y conocer si existe ese personaje en la base de datos de archivo.

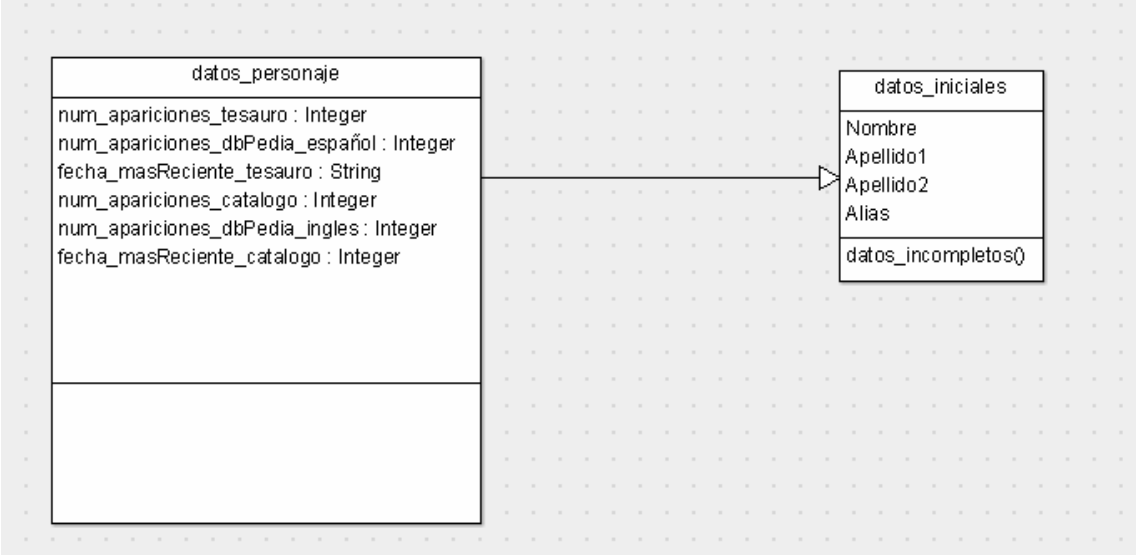

Figura 3.5: Diagrama de clases para el objeto datos_personaje

El segundo diagrama (Fig. 3.6) representa el objeto configuración. Éste contiene todos los datos necesarios para acceder a la base de datos de archivo (asociado al objeto gestor_BD, un objeto obtenido de la empresa, que ya utilizan para acceder a su base de datos), para localizar los archivos XML de configuración proporcionados por el personal informático y para escribir el archivo XML resultante con la ficha del personaje (asociación con el objeto datos_fichero), para dar formato al interfaz de usuario (de entrada y de salida), para realizar consultas a la base de datos de archivo, a DBpedia en español e inglés, y para buscar en Internet otra información como webs personales, blogs, vídeos, imágenes, o redes sociales.

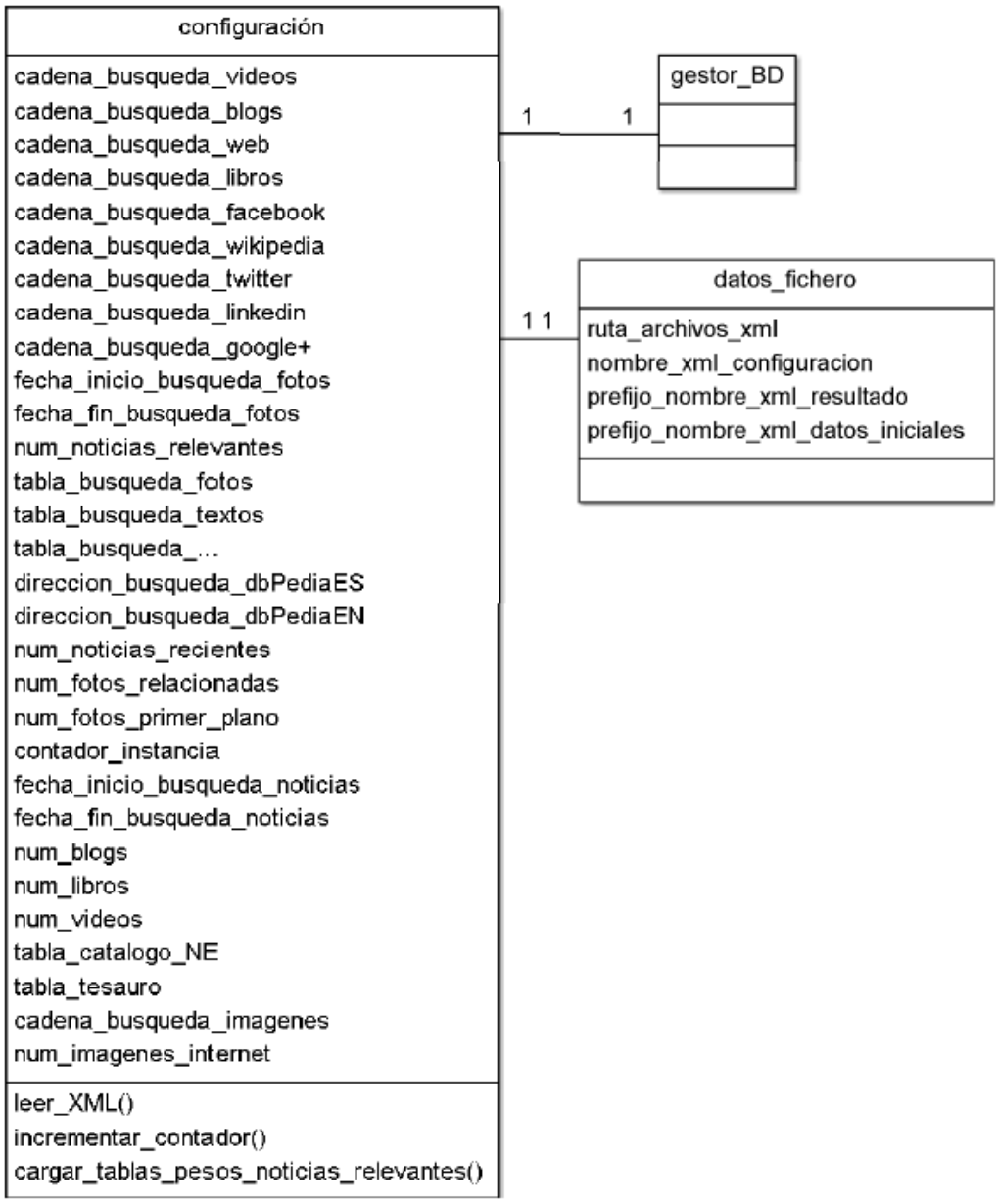

Figura 3.6 Diagrama de clases para el objeto configuración

Finalmente, el tercer diagrama (fig. 3.7) contiene el objeto ficha_personaje que almacena todos los resultados que se mostrarán al usuario a través del interfaz de usuario: Nombre, apellidos, alias, biografía, sucesor, y asociaciones a objetos que completan la ficha: vídeos, libros, imágenes, blogs, redes sociales, webs personales extraídas de Internet, cargos extraídos de DBpedia, noticias y entrevistas de la base de datos de archivo. Todos estos objetos pueden aparecer en un número determinado en la configuración inicial, y contienen los atributos necesarios para localizarlos y extraer bien la imagen o el texto que nos interesa mostrar en la ficha de personaje resultante.

El objeto ficha_personaje contiene también un procedimiento para escribir el archivo XML resultante que utilizará como fuente el interfaz de usuario para mostrar en pantalla.

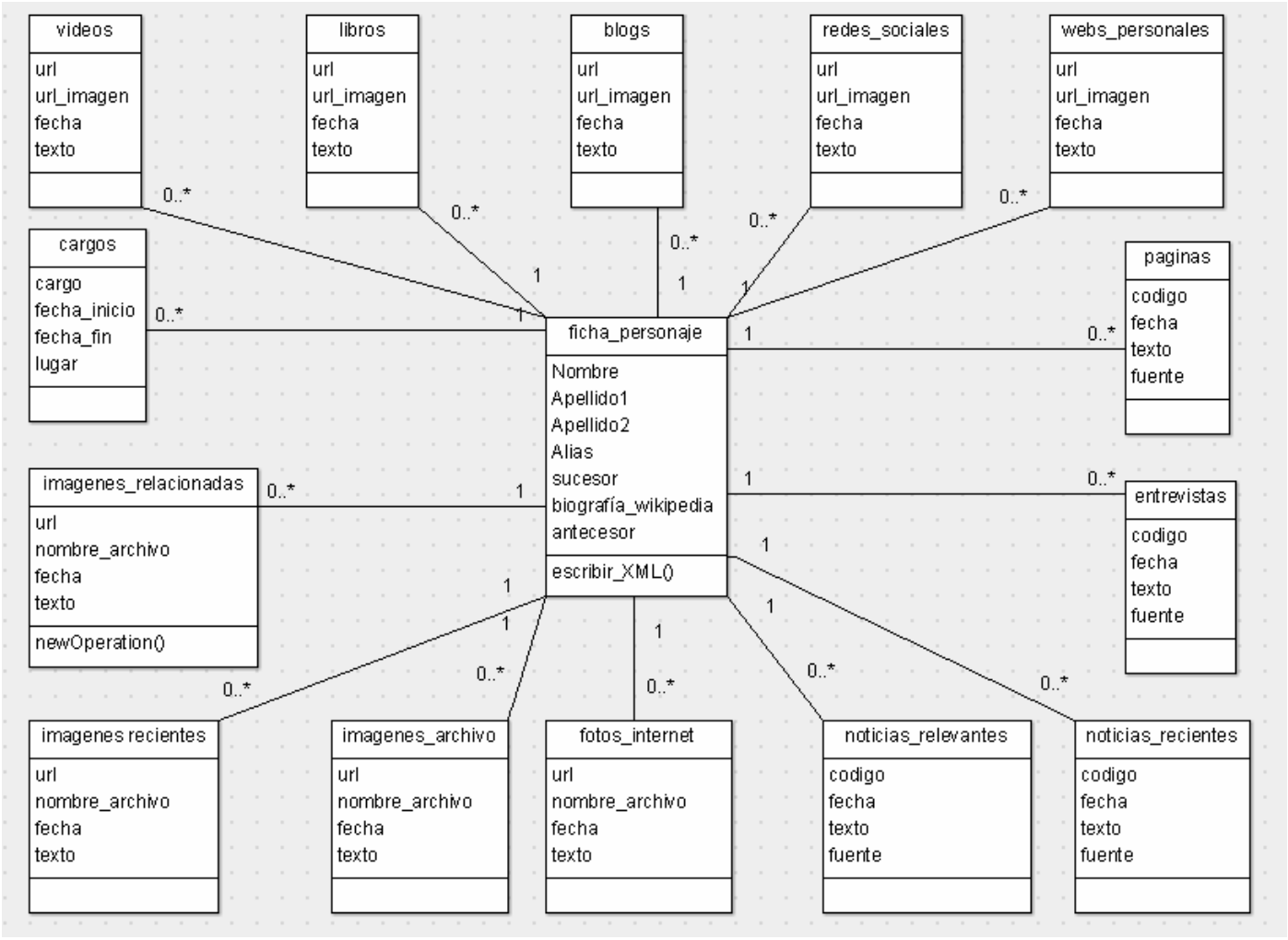

Figura 3.7 Diagrama de clases para el objeto ficha_personaje

B2.4 Casos de uso.

Descripción y diagramas de los casos de uso:

NIVEL 0

En la figura 3.8 vemos los principales actores que intervienen en este sistema: el Personal Documentalista y el Periodista acceden únicamente al interfaz de usuario para solicitar una respuesta, y el Personal de Informática que accede al configurador para proporcionar datos de configuración del sistema.

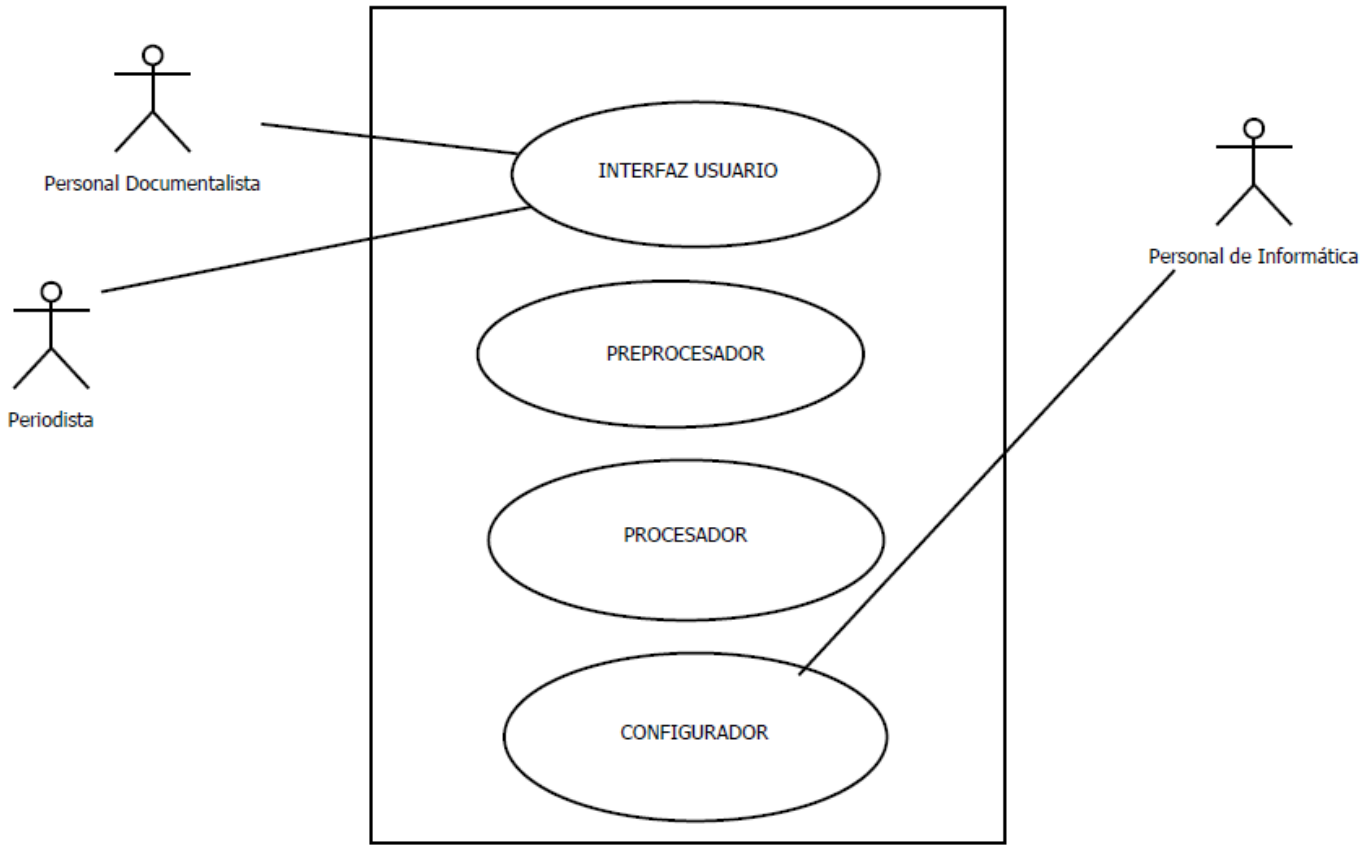

Figura 3.8 Diagrama Caso de uso de nivel 0

NIVEL 1

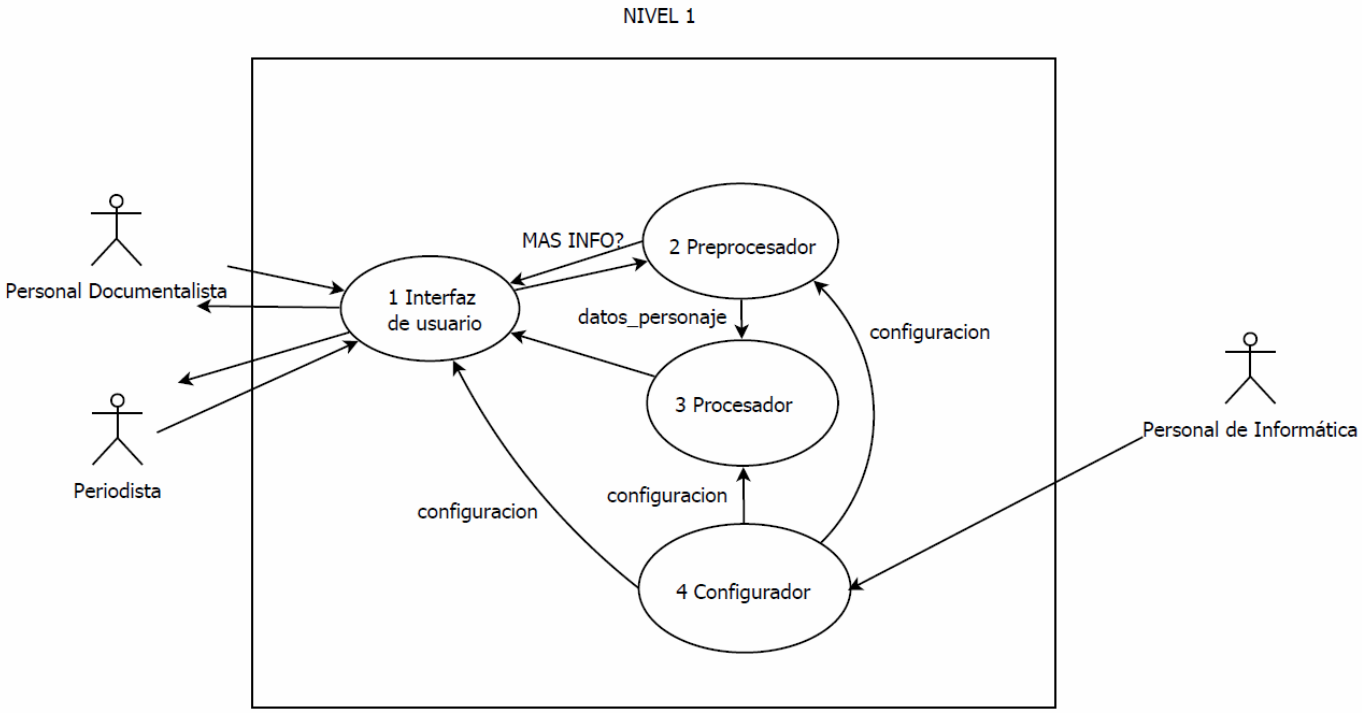

Figura 3.9 Diagrama Caso de uso de nivel 1

Caso de uso:	1. Interfaz de usuario
Propósito:	Recoger y mostrar información al usuario
Actor principal:	Personal documentalista, Periodista
Descripción:	Recoge los datos del personaje que busca el usuario. Muestra la web resultante con la ficha biográfica del personaje
Flujo básico:	Acciones: 1 y 2 . La herramienta de presentación (de documentalista o de periodista) recoge los datos iniciales del personaje. 3. El generador de XML crea con los datos iniciales un archivo XML que envía al preprocesador. 4. Lee los datos de configuración para las herramientas de presentación y el procesador de XML. 5. Procesa el archivo XML que contiene la ficha biográfica y genera un archivo HTML que pasa a las herramientas de presentación. 6. Las herramientas de presentación muestran el resultado al usuario.
Flujo alternativo:	3.A- datos incompletos 3.A- Pedir más información al usuario

Caso de uso:	2. Preprocesador
Propósito:	Determinar el personaje objeto de la búsqueda.
Precondiciones:	El interfaz de usuario nos ha pasado los datos del personaje
Descripción:	Comprueba los datos de entrada. Cuantifica y sitúa en el tiempo las apariciones del personaje en el archivo documental e Internet. Con esos datos desambigua entre varios personajes si es necesario.
Flujo básico:	Acciones: 1. Procesar XML con los datos iniciales del personaje para comprobar si están todos los necesarios para la búsqueda. 2. Buscar el personaje en el archivo documental e Internet para valorar frecuencia y fechas de apariciones en ellos. 3. Si aparecen varios personajes coincidentes con los datos iniciales, desambiguar y escoger uno. 4. Enviar al procesador los datos resultantes de la búsqueda previa correspondientes al personaje escogido.
Flujo alternativo:	1.A- datos incompletos 1.A-Informar al interfaz de usuario de que los datos son incompletos

Caso de uso:	3. Procesador
Propósito:	Buscar la información necesaria para completar la ficha biográfica del personaje
Precondiciones:	El preprocesador nos ha enviado los datos iniciales del personaje correspondientes a la búsqueda previa.
Descripción:	Busca toda la información relativa al personaje en el archivo documental e Internet, y crea un XML con los resultados que será usado por la interfaz de usuario para mostrarlos.
Flujo básico:	Acciones: 1. Busca información en el archivo documental. 2. Busca información en Internet. 3. Busca información en DBpedia. 4. Crea un archivo XML con los resultados de la búsqueda, según los parámetros de configuración.

Caso de uso:	4. Configurador
Propósito:	Configurar los procesos que forman la aplicación.
Actor principal:	Personal informático
Precondiciones:	El personal informático debe ser un usuario autorizado.
Descripción:	Crea un archivo XML que contiene la información que necesitan el interfaz de usuario, el preprocesador y el procesador para realizar sus búsquedas y ajustar los resultados.
Flujo básico:	Acciones: 1. Validación de usuario 2. Escritura del archivo XML con la información del interfaz de usuario, preprocesador y procesador.

NIVEL 2

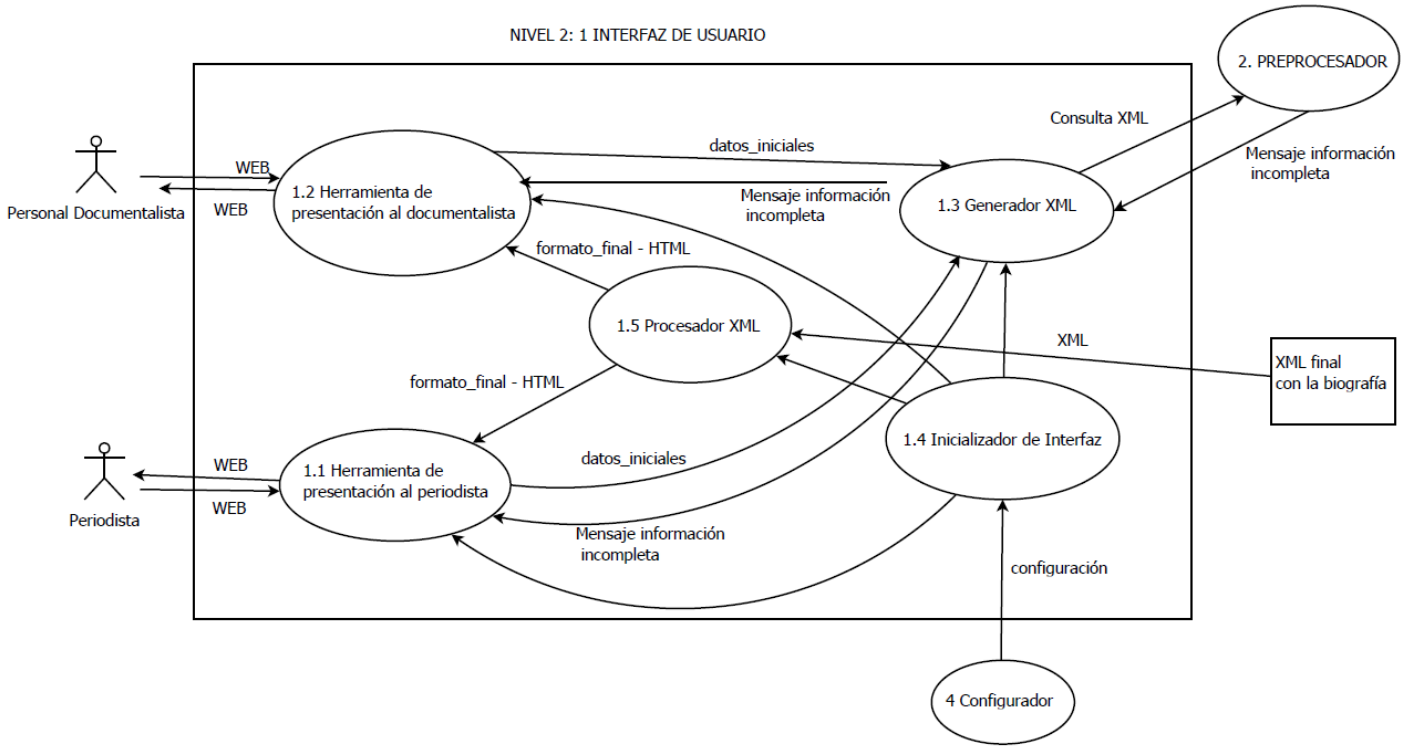

Figura 3.10 Diagrama Caso de uso de nivel 2, Interfaz de usuario.

Caso de uso:	1.1 Herramienta de presentación al periodista
Propósito:	Recoger los datos de entrada del periodista y mostrarle por pantalla la ficha biográfica.
Actor principal:	Periodista
Descripción:	Recoge el nombre, apellidos y alias del personaje que busca el periodista. Lo convierte en un archivo XML y lo envía al preprocesador para comprobar si están presentes todos los datos necesarios. Muestra por pantalla el archivo HTML con la ficha biográfica.
Flujo básico:	1. Solicitar datos al periodista 2. Enviar datos iniciales al generador de XML 3. Si el preprocesador considera que son incompletos, envía mensaje error al periodista. 4. Lee los datos de configuración necesarios para crear las web de recogida y muestra de datos. 5. Muestra la ficha biográfica del personaje.

Caso de uso:	1.2 Herramienta de presentación al documentalista
Propósito:	Recoger los datos de entrada del documentalista y mostrarle por pantalla la ficha biográfica.
Actor principal:	Personal documentalista
Descripción:	Recoge el nombre, apellidos y alias del personaje que busca el documentalista. Lo convierte en un archivo XML y lo envía al preprocesador para comprobar si están presentes todos los datos necesarios. Muestra por pantalla el archivo HTML con la ficha biográfica.
Flujo básico:	1. Solicitar datos al documentalista. 2. Enviar datos iniciales al generador de XML 3. Si el preprocesador considera que son incompletos, envía mensaje error al periodista. 4. Lee los datos de configuración necesarios para crear las web de recogida y muestra de datos. 5. Muestra la ficha biográfica del personaje.

Caso de uso:	1.3 Generador de XML
Propósito:	Enviar datos iniciales al preprocesador.
Precondiciones:	Que el periodista o el documentalista hayan completado la entrada de datos a través de las herramientas de presentación.
Descripción:	Convierte los datos introducidos en la web de presentación en un archivo XML que leerá el preprocesador.
Flujo básico:	1. Escribe archivo XML con los datos recibidos de las herramientas de presentación. 2. Si recibe mensaje de información incompleta, se lo envía a las herramientas de presentación.

Caso de uso:	1.4 Inicializador de interfaz
Propósito:	Pasar los datos de configuración necesarios para la presentación a las herramientas correspondientes, y al generador de XML para crear el fichero final.
Precondiciones:	Necesita la configuración proporcionada por el personal informático.
Descripción:	Lee la configuración desde el archivo XML creado por el configurador y envía los datos necesarios a la herramienta de presentación para periodistas y a la herramienta de presentación para documentalistas. También informa al generador de XML y al procesador de XML de los formatos que tienen que utilizar.
Flujo básico:	1. Lee el archivo XML creado desde el configurador. 2. Envía los datos necesarios a las dos herramientas de presentación para que puedan crear la pantalla que presentará al usuario la ficha biográfica. 3. Envía al procesador de XML la información necesaria para transformar el XML final en un HTML. 4. Envía la información necesaria al generador XML para que escriba el archivo XML con los datos iniciales del personaje.

Caso de uso:	1.5 Procesador de XML
Propósito:	Leer el archivo XML con la ficha biográfica y transformarlo en un archivo HTML.
Precondiciones:	Que exista el archivo XML final.
Descripción:	Lee el archivo XML final, y junto con la información recibida del inicializador de interfaz, crea un archivo HTML que utilizarán las herramientas de presentación para los usuarios finales.
Flujo básico:	1. Recibe los datos de configuración 2. Lee el archivo XML con la ficha biográfica 3. Crea el archivo HTML para el periodista o para el documentalista, según haya sido la solicitud.

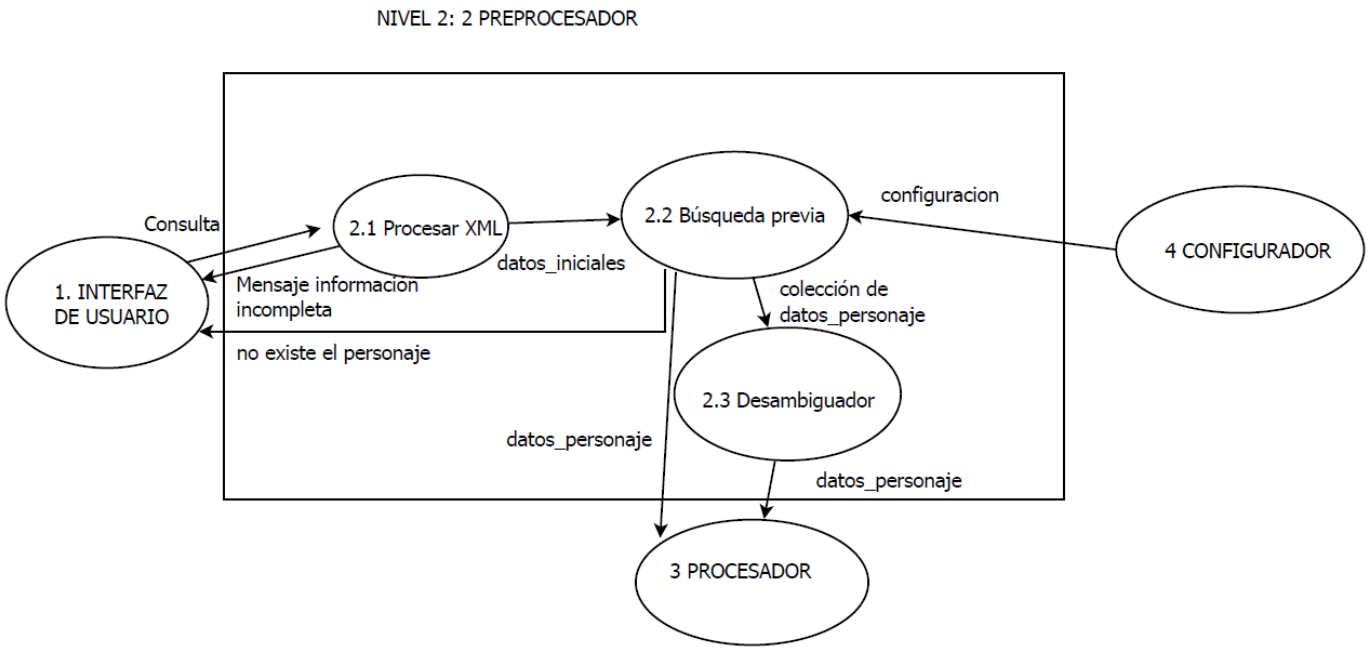

Figura 3.11 Diagrama Caso de uso de nivel 2, Preprocesador.

Caso de uso:	2.1 Procesador de XML
Propósito:	Leer el XML con los datos iniciales del personaje, y ver si están todos los necesarios para comenzar la búsqueda.
Precondiciones:	Debe existir el XML de datos iniciales creado por el interfaz de usuario.
Descripción:	Recibe la información de que existe un XML, lo lee y comprueba que estén los datos necesarios para realizar la búsqueda previa. Si no lo están, envía mensaje de error al interfaz de usuario. En caso de que sean correctos, pasa los datos iniciales al proceso de búsqueda previa. Si el proceso de búsqueda previa detecta que no existe ningún personaje que coincida con los datos iniciales, este proceso también devuelve un mensaje de aviso para la interfaz de usuario.
Flujo básico:	1. Recibe aviso de creación del XML 2. Lee XML 3. Comprueba que los datos sean los necesarios 4. Envía datos iniciales a búsqueda previa
Flujo alternativo:	3.A Si los datos no están completos - Envía mensaje de error al interfaz de usuario.

Caso de uso:	2.2 Búsqueda previa
Propósito:	Realizar un primer análisis de la presencia del personaje tanto en la archivo documental como en Internet.
Precondiciones:	Necesita que el procesador de XML le pase los datos iniciales del personaje.
Descripción:	Busca y cuenta apariciones del personaje en el tesauro, en el catálogo de Named Entities, en DBpedia español e inglés. También almacena las fechas de las apariciones. Devuelve un personaje, o en caso de ambigüedad, una colección de personajes para el desambiguador.
Flujo básico:	1. Busca el personaje en el tesauro, cuenta y almacena fechas. 2. Busca el personaje en el catálogo de Named Entities. Cuenta y almacena fechas de aparición. 3. Busca el personaje en DBpedia en español. 4. Busca el personaje en DBpedia en inglés. 5. Si sólo se ha encontrado un posible personaje, se pasan los datos al procesador.
Flujo alternativo:	5.A Si no existe el personaje - Envía un mensaje de aviso para la interfaz de usuario. 5.B Si hay diferentes personajes encontrados con los mismos datos iniciales en alguno de los pasos - Envía la colección de personajes al desambiguador para que seleccione uno.

Caso de uso:	2.3 Desambiguador
Propósito:	Decidir entre varios personajes, cuál es el más adecuado para hacer su ficha biográfica.
Precondiciones:	Que hayan aparecido diferentes personajes con los mismos datos iniciales en la búsqueda previa.
Descripción:	Decide qué personaje escogemos para crear la ficha biográfica, en función del número de apariciones y fechas, y sus datos en DBpedia.
Flujo básico:	1. Recibe la colección de personajes, y escoge el más frecuente en apariciones, incluyendo la aparición en DBpedia español e inglés. 2. En caso de empate, escoge el más reciente en fechas de aparición.

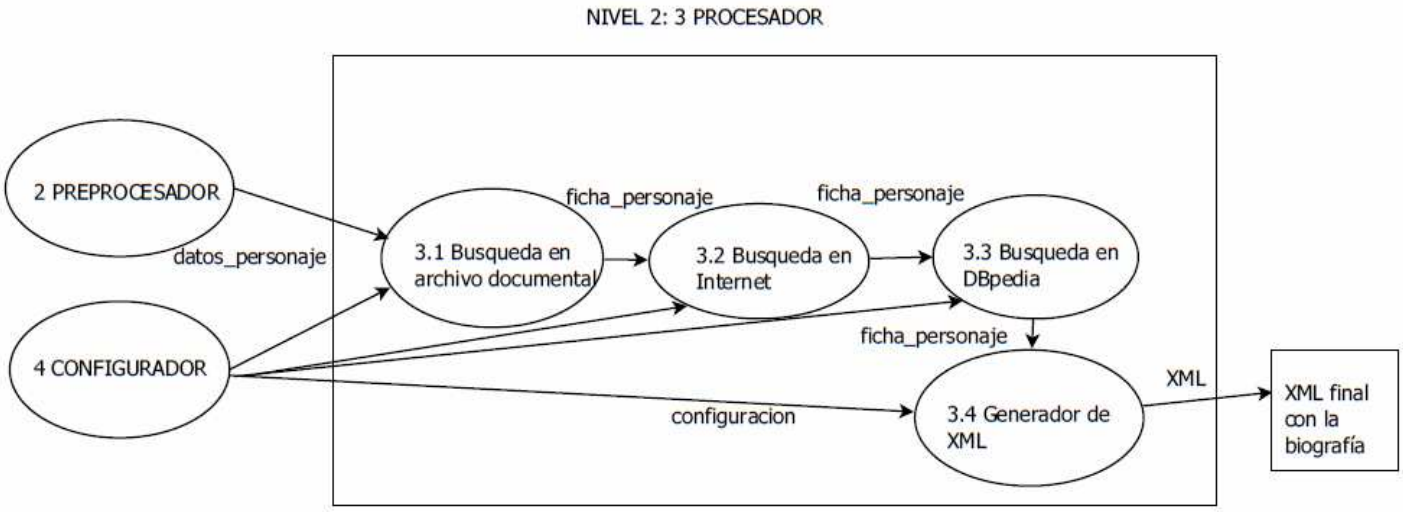

Figura 3.12 Diagrama Caso de uso de nivel 2, Procesador.

Caso de uso:	3.1 Búsqueda en Archivo Documental
Propósito:	Buscar la información necesaria para la ficha biográfica del personaje en la archivo documental.
Precondiciones:	Que el preprocesador pase los datos definitivos sobre el personaje para su búsqueda.
Descripción:	Busca en la archivo documental los artículos y fotografías que vamos a incluir en la ficha biográfica: noticias más recientes, noticias más relevantes, fotografías más recientes, y fotografías de primer plano. Todos los resultados los va añadiendo a la ficha del personaje.
Flujo básico:	1. Búsqueda en BD archivo de las noticias más recientes, según las directrices que marque el configurador respecto al número de noticias y los datos de búsqueda. 2. Búsqueda en BD archivo de las noticias más relevantes, según las directrices que marque el configurador respecto al número de noticias y los datos de búsqueda. 3. Búsqueda en BD archivo de las fotografías más recientes, según las directrices que marque el configurador respecto al número de fotos y los datos de búsqueda. 4. Búsqueda en BD archivo de las fotografías de primer plano, según las directrices que marque el configurador respecto al número de fotos y los datos de búsqueda. 5. Añadir todos los datos encontrados a la ficha del personaje.

Caso de uso:	3.2 Búsqueda en Internet
Propósito:	Buscar la información necesaria para la ficha biográfica del personaje en Internet.
Precondiciones:	Que el preprocesador pase los datos definitivos sobre el personaje para su búsqueda.
Descripción:	Busca en Internet: información en wikipedia, recupera para la ficha imágenes, vídeos, libros, blogs, webs personales y redes sociales. Todos los resultados los va añadiendo a la ficha del personaje.
Flujo básico:	1. Recupera la información en wikipedia. 2. Recupera imágenes del personaje. 3. Recupera vídeos del personaje. 4. Recupera libros del personaje. 5. Recupera blogs del personaje. 6. Recupera redes sociales donde aparece el personaje. 7. Recupera webs personales que posea el personaje. 8. Añadir todos los datos encontrados a la ficha del personaje.

Caso de uso:	3.3 Búsqueda en DBpedia
Propósito:	Buscar información necesaria para la ficha biográfica en DBpedia en español y en inglés.
Precondiciones:	Que el preprocesador pase los datos definitivos sobre el personaje para su búsqueda.
Descripción:	Busca en Internet información en DBpedia, y recupera para la ficha datos biográficos.
Flujo básico:	1. Recuperación de la información en DBpedia español, utilizando los parámetros que nos proporciona el configurador para realizar la búsqueda. 2. Recuperación de la información en DBpedia inglés, utilizando los parámetros que nos proporciona el configurador para realizar la búsqueda. 3. Añadir todos los datos encontrados a la ficha del personaje.

Caso de uso:	3.4 Generador de XML
Propósito:	Elaborar el archivo XML según los parámetros de configuración donde se incluyen todos los datos que componen la ficha biográfica.
Precondiciones:	Que se haya completado la búsqueda de información sobre el personaje.
Descripción:	Recibe los parámetros de configuración desde el configurador, y recibe la información recuperada sobre el personaje. Con todo ello escribe el archivo XML que utilizará el interfaz de usuario para mostrar los resultados.
Flujo básico:	1. Escribe archivo XML según los datos de configuración y del personaje.

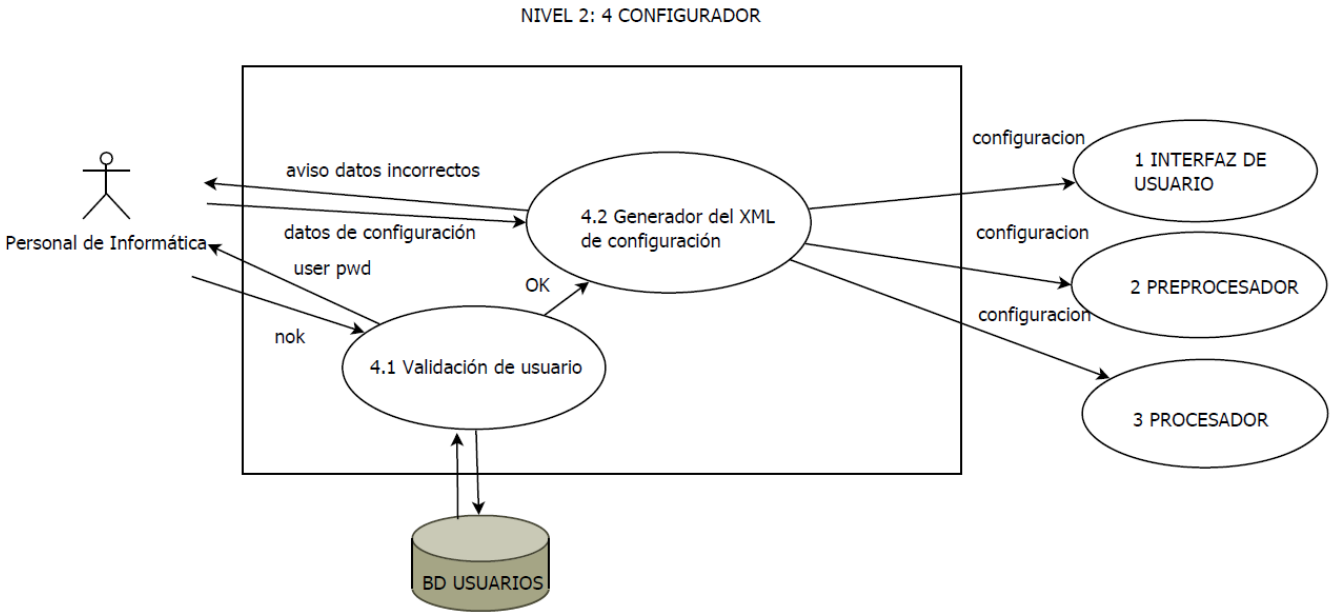

Figura 3.13 Diagrama Caso de uso de nivel 2, Configurador

Caso de uso:	4.1 Validación de usuario
Propósito:	Conocer si el usuario es un personal de informática autorizado para configurar la aplicación.
Actor principal:	Personal de informática
Descripción:	Recibe el nombre y password de usuario, y consulta contra la base de datos de usuarios si está autorizado para el uso o no de la aplicación.
Flujo básico:	1. Lectura de datos de usuario. 2. Consulta a la base de datos de usuarios. 3. Da acceso al proceso que genera el XML de configuración.
Flujo alternativo:	2.A-Si no está autorizado devuelve mensaje de error.

Caso de uso:	4.2 Generador del XML de configuración
Propósito:	Crear un archivo XML que contenga la información para configurar los diferentes procesos.
Descripción:	Permite al usuario introducir los datos que necesita el XML de configuración.
Flujo básico:	1. Recibe los datos por parte del usuario. 2. Comprueba dichos datos. 3. Genera un archivo XML en base a esos datos.
Flujo alternativo:	3.A – En caso de que algún dato no sea válido -Mostrar un aviso al usuario.

NIVEL 3

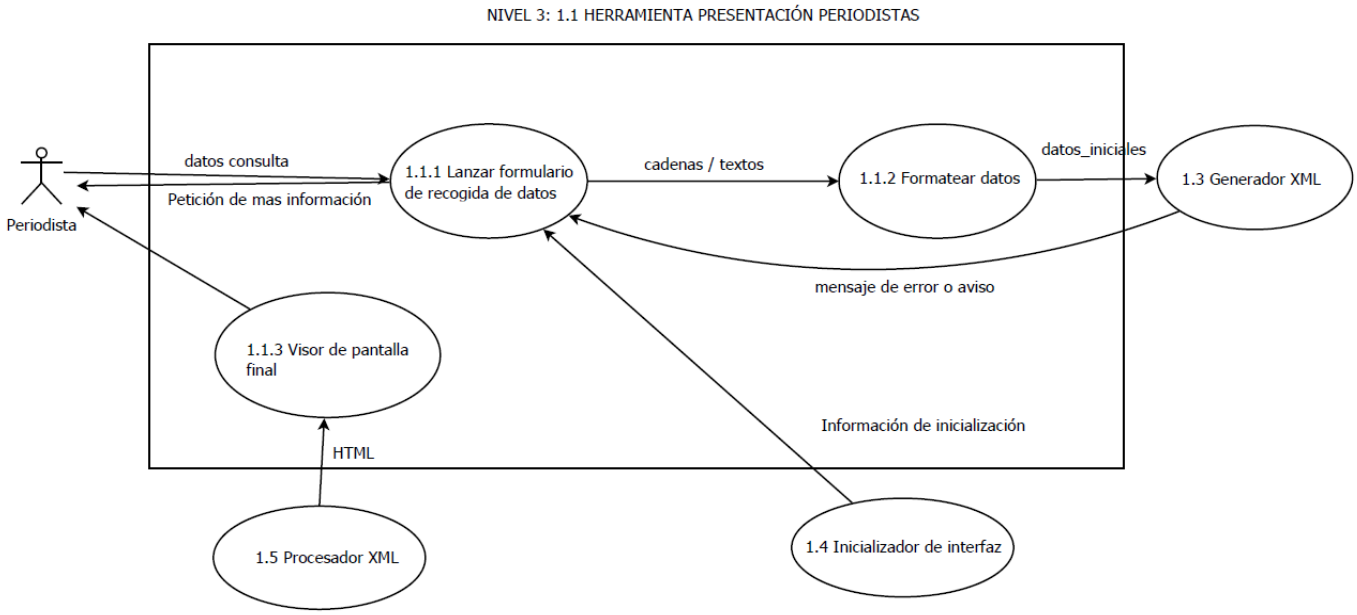

Figura 3.14 Diagrama Caso de uso de nivel 3, Herramienta presentación Periodistas.

Caso de uso:	1.1.1 Lanzar formulario de recogida de datos
Propósito:	Mostrar en pantalla un formulario que recoja los datos necesarios sobre el personaje del que queremos crear la ficha biográfica.
Actor principal:	Periodista
Precondiciones:	Recibe la información de formato de la pantalla desde el Inicializador de interfaz.
Descripción:	Lanza el formulario de recogida de datos en función de los datos de inicialización, y pasa la información introducida por el usuario al proceso de formateo de datos. Avisa al usuario de falta de datos o la no existencia del personaje.
Flujo básico:	1. Lanza el formulario según los parámetros de inicialización. 2. Lee los datos y los envía al proceso de formatear datos.
Flujo alternativo:	Si el proceso generador de XML devuelve un mensaje de error o aviso -Lo muestra al usuario.

Caso de uso:	1.1.2 Formatear datos
Propósito:	Recibir los datos recogidos del usuario y darles formato para el uso posterior del proceso 1.3 Generador de XML
Precondiciones:	El usuario ha introducido los datos del personaje a buscar.
Descripción:	Recibe los textos recogidos en el formulario de entrada y les da un formato coherente para enviarlos al proceso posterior.
Flujo básico:	1. Lectura de los datos recibidos. 2. Formateo de los datos.

Caso de uso:	1.1.3 Visor de pantalla final
Propósito:	Mostrar en pantalla los datos encontrados sobre el personaje del que hemos solicitado la ficha biográfica.
Actor principal:	Periodista
Precondiciones:	Recepción del archivo HTML con el resultado de la búsqueda.
Descripción:	Lee el archivo HTML con la ficha biográfica y la muestra en pantalla al usuario.
Flujo básico:	1. Leer archivo HTML 2. Mostrar resultados contenidos en el archivo en pantalla.

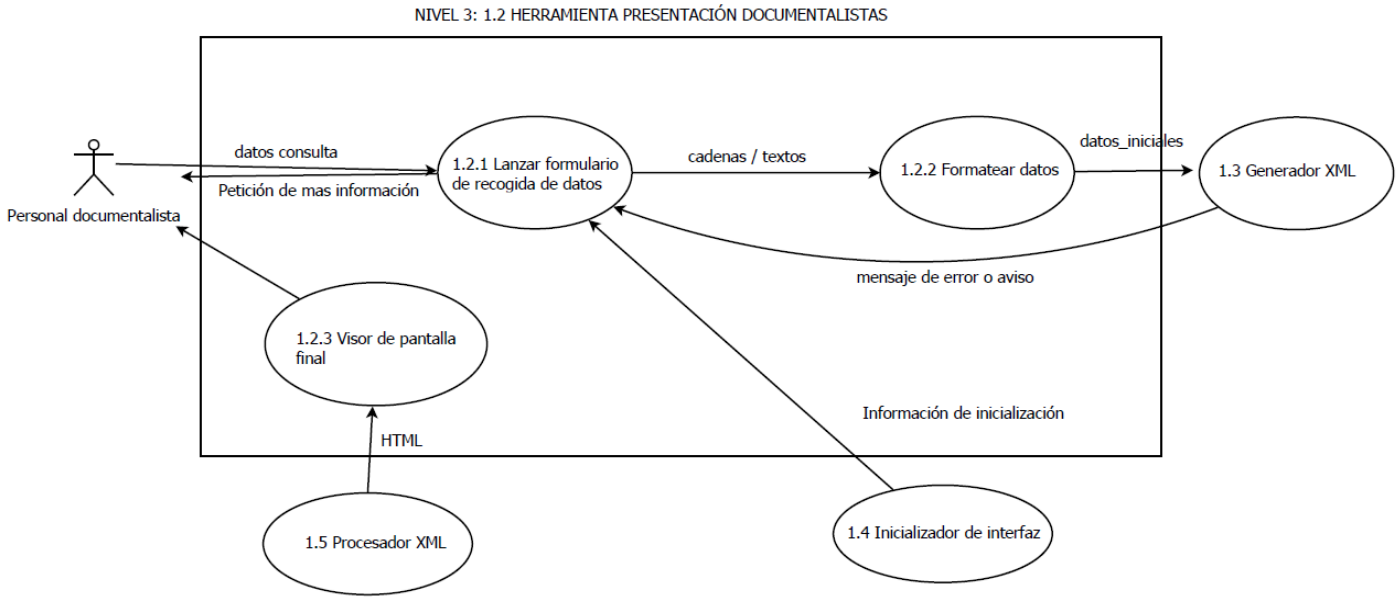

Figura 3.15 Diagrama Caso de uso de nivel 3, Herramienta presentación Documentalistas.

Caso de uso:	1.2.1 Lanzar formulario de recogida de datos
Propósito:	Mostrar en pantalla un formulario que recoja los datos necesarios sobre el personaje del que queremos crear la ficha biográfica.
Actor principal:	Personal documentalista
Precondiciones:	Recibe la información de formato de la pantalla desde el Inicializador de interfaz.
Descripción:	Lanza el formulario de recogida de datos en función de los datos de inicialización, y pasa la información introducida por el usuario al proceso de formateo de datos. Avisa al usuario de falta de datos o la no existencia del personaje.
Flujo básico:	1. Lanza el formulario según los parámetros de inicialización. 2. Lee los datos y los envía al proceso de formatear datos.
Flujo alternativo:	Si el proceso generador de XML devuelve un mensaje de error o aviso -Lo muestra al usuario.

Caso de uso:	1.2.2 Formatear datos
Propósito:	Recibir los datos recogidos del usuario y darles formato para el uso posterior del proceso 1.3 Generador de XML
Precondiciones:	El usuario ha introducido los datos del personaje a buscar.
Descripción:	Recibe los textos recogidos en el formulario de entrada y les da un formato coherente para enviarlos al proceso posterior.
Flujo básico:	1. Lectura de los datos recibidos. 2. Formateo de los datos.

Caso de uso:	1.2.3 Visor de pantalla final
Propósito:	Mostrar en pantalla los datos encontrados sobre el personaje del que hemos solicitado la ficha biográfica.
Actor principal:	Personal documentalista
Precondiciones:	Recepción del archivo HTML con el resultado de la búsqueda.
Descripción:	Lee el archivo HTML con la ficha biográfica y la muestra en pantalla al usuario.
Flujo básico:	1. Leer archivo HTML 2. Mostrar resultados contenidos en el archivo en pantalla.

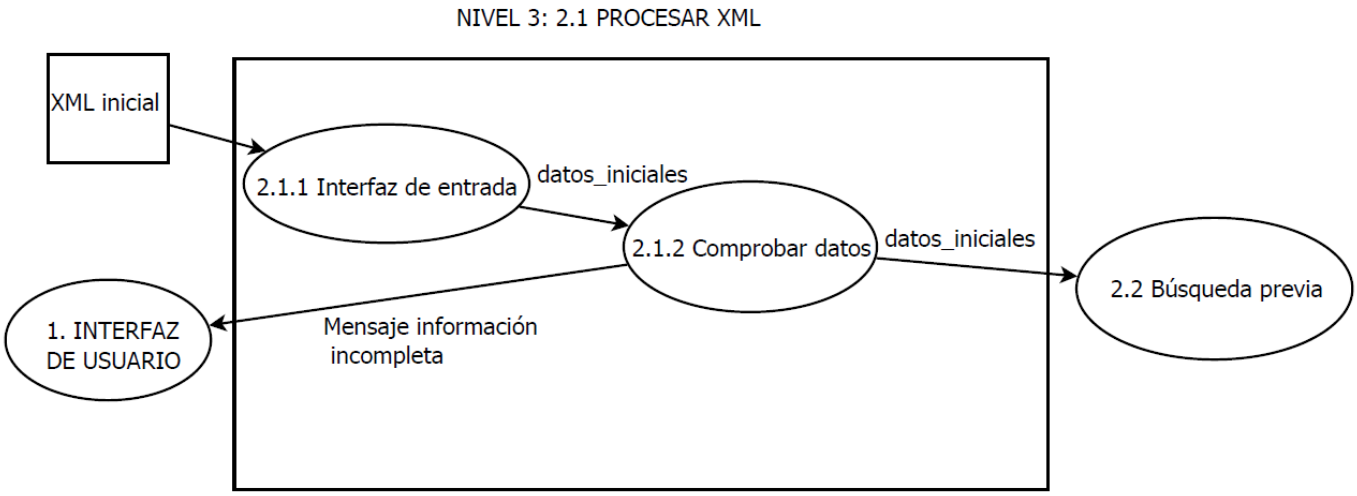

Figura 3.16 Diagrama Caso de uso de nivel 3, Procesar XML.

Caso de uso:	2.1.1 Interfaz de entrada
Propósito:	Leer el archivo XML con los datos del personaje introducidos por el usuario.
Precondiciones:	El archivo XML debe existir previamente.
Descripción:	Lee los datos introducidos por el usuario contenidos en el XML inicial, y los carga en memoria para su uso posterior por parte de otros procesos.
Flujo básico:	1. Leer archivo XML. 2. Cargar en memoria los datos leídos del archivo.

Caso de uso:	2.1.2 Comprobar datos
Propósito:	Verificar si el usuario ha introducido los datos necesarios sobre el personaje para comenzar la búsqueda.
Precondiciones:	El usuario ha introducido los datos y éstos han sido leídos por el proceso 2.1.1.
Descripción:	Verifica, según una lista interna de requisitos, si recibe todos los datos necesarios para comenzar la búsqueda para la ficha biográfica.
Flujo básico:	1. Comprobación de la existencia de los datos imprescindibles para la búsqueda. 2. Pasar los datos al siguiente proceso de búsqueda previa.
Flujo alternativo:	1.A- En caso de que falte alguno, devolver mensaje de error.

Caso de uso:	2.2.1 Búsqueda en tesauro
Propósito:	Comprobar la presencia del personaje en el tesauro del archivo documental.
Descripción:	Recibe los datos iniciales introducidos por el usuario y busca con ellos apariciones del personaje en el tesauro del archivo documental. Almacena y cuenta las apariciones y fechas de cada personaje encontrado que se corresponda con dichos datos iniciales.
Flujo básico:	1. Lee datos iniciales del personaje y de configuración. 2. Realiza la consulta buscando apariciones del personaje en el tesauro del archivo documental. 3. Cuenta y almacena cada aparición del personaje y guarda su fecha más reciente.

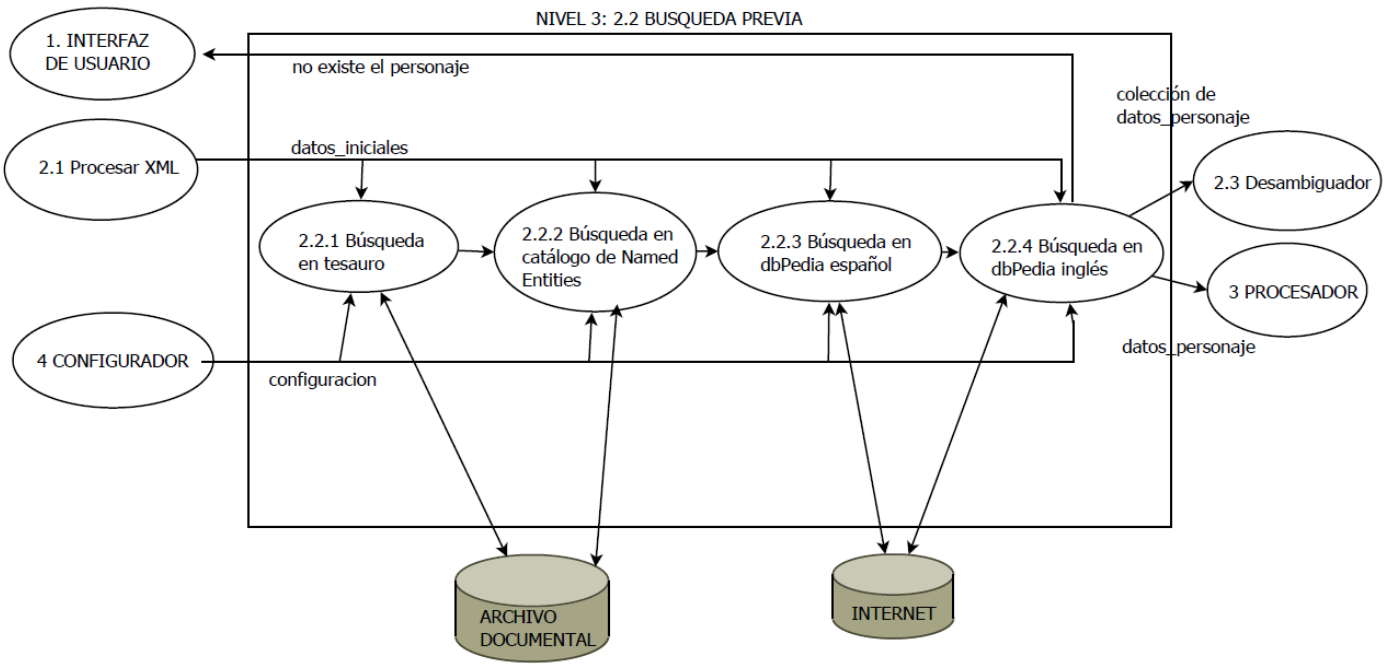

Figura 3.17 Diagrama Caso de uso de nivel 3, Búsqueda previa.

Caso de uso:	2.2.2 Búsqueda en catálogo de Named Entities
Propósito:	Comprobar la presencia del personaje en el catálogo de Named Entities.
Precondiciones:	Hemos creado previamente un catálogo de Named Entities donde realizar la búsqueda.
Descripción:	Recibe los datos introducidos por el usuario y busca con ellos apariciones del personaje en el catálogo de Named Entities. Almacena y cuenta las apariciones y fechas de cada personaje encontrado que se corresponda con dichos datos iniciales.
Flujo básico:	1. Lee datos iniciales del personaje y de configuración. 2. Realiza la consulta buscando apariciones del personaje en el catálogo de Named Entities del archivo documental. 3. Cuenta y almacena cada aparición del personaje y guarda su fecha más reciente.

Caso de uso:	2.2.3 Búsqueda en DBpedia español
Propósito:	Comprobar si existe el personaje en DBpedia español.
Descripción:	Recibe los datos introducidos por el usuario y busca con ellos apariciones del personaje en el DBpedia español. Almacena y cuenta las apariciones de cada personaje encontrado que se corresponda con dichos datos iniciales.
Flujo básico:	1. Lee datos iniciales del personaje. 2. Realiza la consulta buscando apariciones del personaje en el DBpedia español. 3. Cuenta y almacena cada aparición del personaje.

Caso de uso:	2.2.4 Búsqueda en DBpedia inglés
Propósito:	Comprobar si existe el personaje en DBpedia inglés.
Descripción:	Recibe los datos introducidos por el usuario y busca con ellos apariciones del personaje en el DBpedia inglés. Almacena y cuenta las apariciones de cada personaje encontrado que se corresponda con dichos datos iniciales.
Flujo básico:	1. Lee datos iniciales del personaje. 2. Realiza la consulta buscando apariciones del personaje en el DBpedia inglés. 3. Cuenta y almacena cada aparición del personaje.

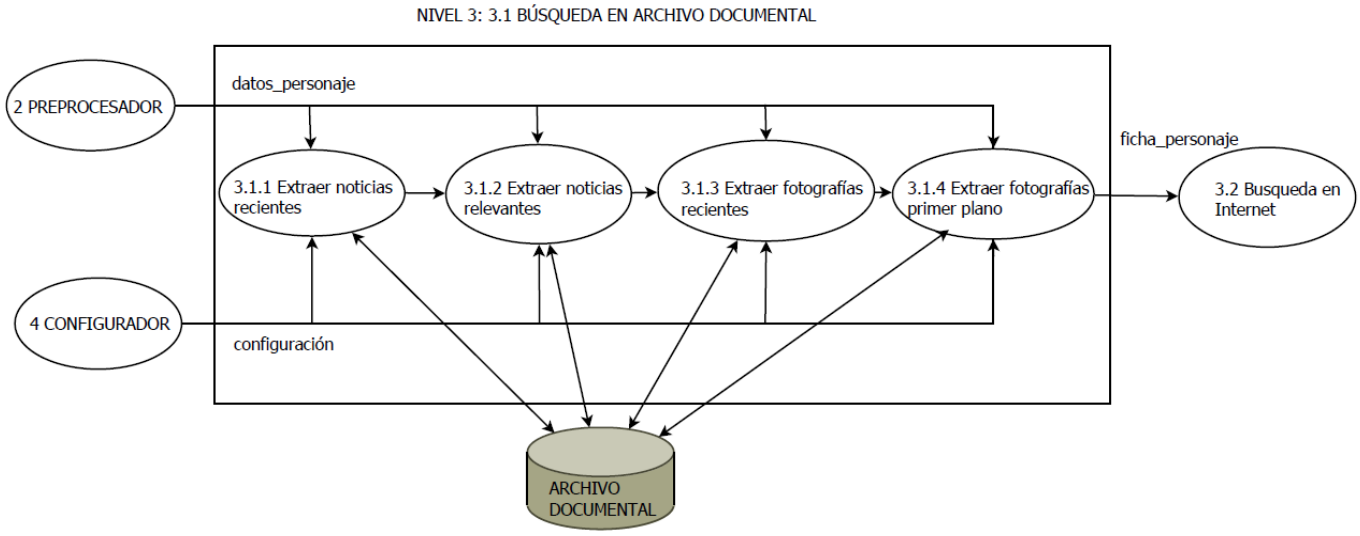

Figura 3.18 Diagrama Caso de uso de nivel 3, Búsqueda en archivo documental.

Caso de uso:	3.1.1 Extraer noticias recientes
Propósito:	Buscar las noticias con fecha más reciente sobre el personaje.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Consulta al archivo documental, y extrae de él, según los límites de fechas y el número máximo de noticias que marca el configurador, las noticias más recientes que nombran al personaje buscado.
Flujo básico:	1. Lectura de datos del personaje. 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Consulta en el archivo documental. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados de la consulta.

Caso de uso:	3.1.2 Extraer noticias relevantes
Propósito:	Buscar las noticias con mayor importancia publicadas sobre el personaje.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Consulta al archivo documental, y extrae de él, según los límites de fechas y el número máximo de noticias que marca el configurador, las noticias más relevantes que nombran al personaje buscado.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Consulta al archivo documental. 4. Filtrado de los resultados según los parámetros del configurador y el algoritmo que valora la relevancia de cada resultado. 5. Almacenamiento en memoria de los resultados de la consulta.

Caso de uso:	3.1.3 Extraer fotografías recientes
Propósito:	Buscar las fotografías con fechas más recientes sobre el personaje.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Consulta al archivo documental, y extrae de él, según los límites de fechas y el número máximo de fotografías que marca el configurador, las fotografías más recientes que muestran al personaje buscado.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Consulta al archivo documental. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados de la consulta.

Caso de uso:	3.1.4 Extraer fotografías primer plano
Propósito:	Buscar las fotografías de primer plano almacenadas sobre el personaje.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae del archivo documental, y según los parámetros del configurador, las fotografías de primer plano que muestran al personaje buscado.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Consulta al archivo documental. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados de la consulta.

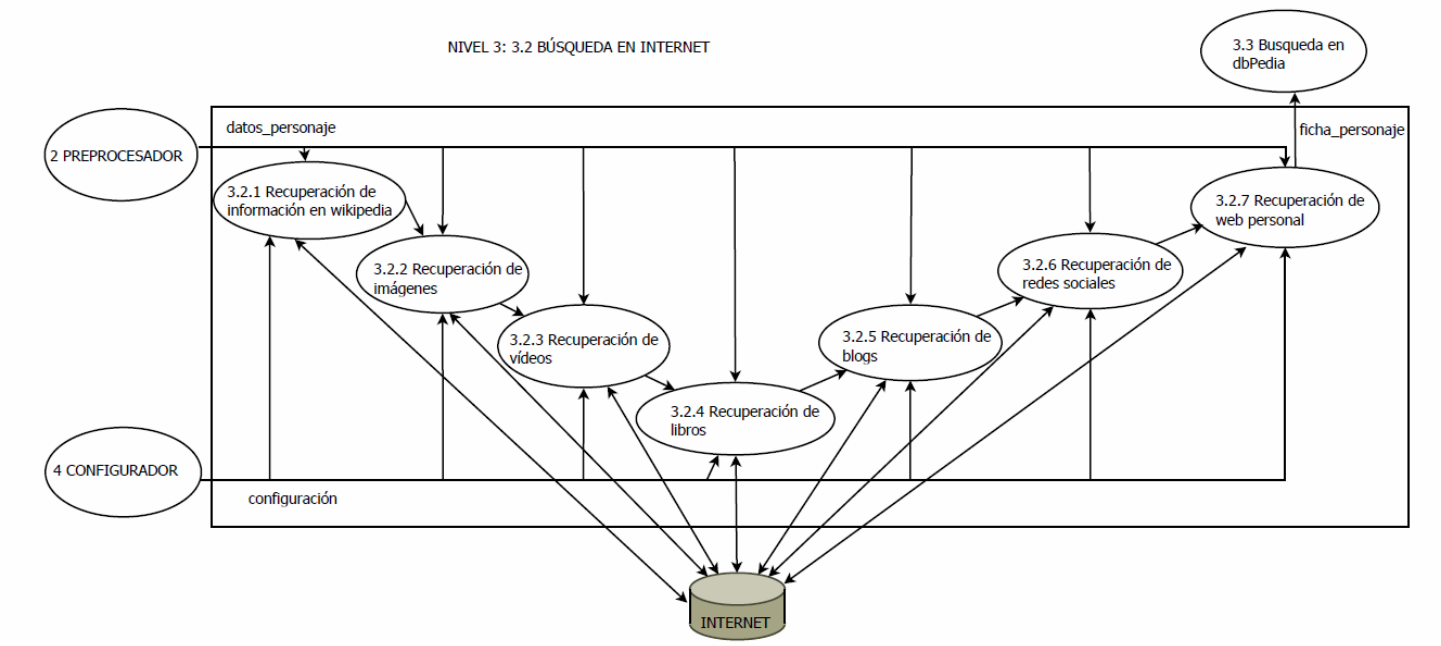

Figura 3.19 Diagrama Caso de uso de nivel 3, Búsqueda en Internet.

Caso de uso:	3.2.1 Recuperación de información en Wikipedia
Propósito:	Localizar al personaje en la web de Wikipedia, para extraer de ella información que complete la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de la página de wikipedia, y según los parámetros del configurador, la información que nos completa la extraída anteriormente del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización y lectura de la web de wikipedia correspondiente al personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

Caso de uso:	3.2.2 Recuperación de imágenes
Propósito:	Localizar imágenes del personaje en la web, para completar la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de Internet, y según los parámetros del configurador, las fotografías que nos completan las extraídas del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización de fotografías en la web correspondientes al personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

Caso de uso:	3.2.3 Recuperación de vídeos
Propósito:	Localizar vídeos del personaje en la web, para completar la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de Internet, y según los parámetros del configurador, los vídeos que nos completan la información extraída del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización de vídeos en la web correspondientes al personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

Caso de uso:	3.2.4 Recuperación de libros
Propósito:	Localizar libros sobre el personaje en la web, para completar la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de Internet, y según los parámetros del configurador, los libros que nos completan la información extraída del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización de libros en la web correspondientes al personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

Caso de uso:	3.2.5 Recuperación de blogs
Propósito:	Localizar blogs del personaje en la web, para completar la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de Internet, y según los parámetros del configurador, los blogs que nos completan la información extraída del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización de blogs en la web correspondientes al personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

Caso de uso:	3.2.6 Recuperación de webs personales
Propósito:	Localizar webs del personaje en Internet, para completar la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de Internet, y según los parámetros del configurador, las webs dedicadas al personaje que nos completan la información extraída del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización de webs en Internet correspondientes al personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

Caso de uso:	3.2.7 Recuperación de redes sociales
Propósito:	Localizar redes sociales en las que participa el personaje, para completar la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de Internet, y según los parámetros del configurador, las apariciones del personaje en las redes sociales que nos completan la información extraída del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización de redes sociales en la web en las que participa el personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

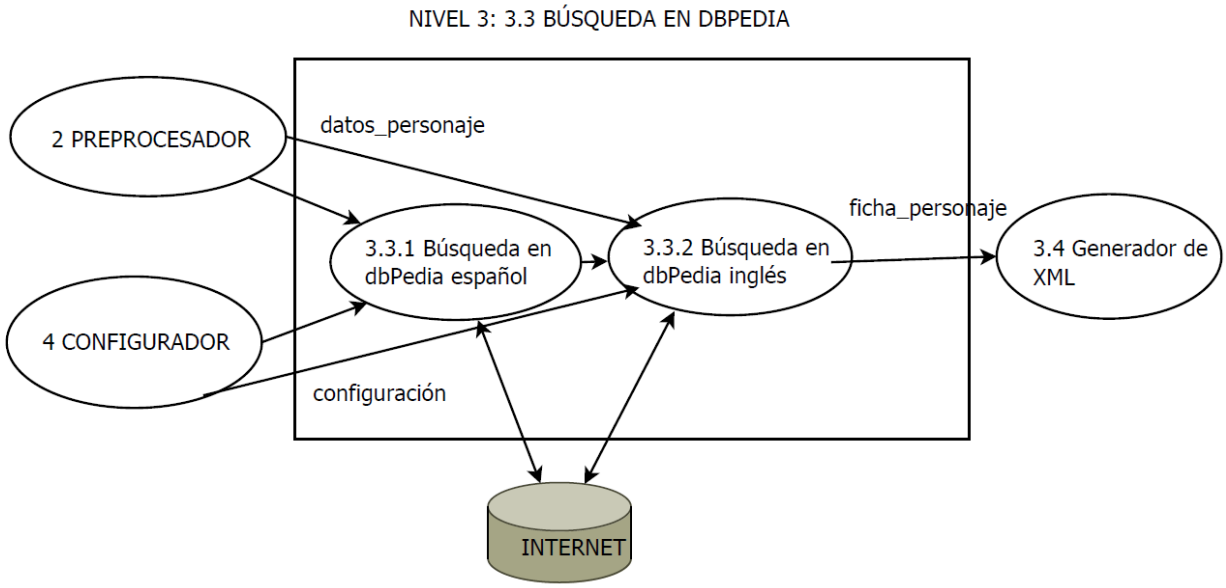

Figura 3.20 Diagrama Caso de uso de nivel 3, Búsqueda en DBpedia.

Caso de uso:	3.3.1 Búsqueda en DBpedia español
Propósito:	Localizar datos del personaje en DBpedia español, para completar la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de la página de DBpedia español, y según los parámetros del configurador, la información que nos completa la extraída anteriormente del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización y lectura de la web de DBpedia español correspondiente al personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

Caso de uso:	3.3.2 Búsqueda en DBpedia inglés
Propósito:	Localizar datos del personaje en DBpedia inglés, para completar la ficha biográfica.
Precondiciones:	Tenemos ya un solo personaje (no hay ambigüedad) con sus datos completos.
Descripción:	Extrae de la página de DBpedia inglés, y según los parámetros del configurador, la información que nos completa la extraída anteriormente del archivo documental.
Flujo básico:	1. Lectura de datos del personaje 2. Lectura de todos los parámetros necesarios enviados desde el configurador. 3. Localización y lectura de la web de DBpedia inglés correspondiente al personaje. 4. Filtrado de los resultados según los parámetros del configurador. 5. Almacenamiento en memoria de los resultados.

B2.5 Base de datos

Para modelar las necesidades del sistema respecto al almacenamiento de la información, se incluye aquí el diagrama entidad-relación. Hay que recordar que este proyecto utiliza una gran base de datos que es EMMA. Aquí se detalla únicamente la parte que afecta a este proyecto. En este PFC el trabajo relativo a bases de datos es principalmente de acceso a ella, ya que sólo se han creado dos entidades nuevas (el catálogo de Named Entities, y el ranking de valoración de noticias) que se han incorporado a la BD ya existente en el periódico.

Sección de la base de datos EMMA relacionada con los textos.

En primer lugar se incluye un fragmento del diagrama E-R que modela el almacenamiento de los textos en los que buscamos la información (Fig. 3.21). Este esquema es un modelo simplificado de las entidades que ya existen y a las que accede el generador de fichas biográficas. Este diagrama está extraído del modelo general del periódico, y no se ha creado para este PFC, sólo se incluye porque para elaborar la herramienta se tiene que conocer este modelo.

Textos. La entidad TEXTOS es el elemento principal, donde se almacena el texto de las noticias publicadas. A cada noticia le corresponderá un resumen, una serie de palabras clave que definen el contenido de la noticia, el autor o autores, la fecha de publicación, el título, el pie de foto (si lo tiene). También se registra el número de palabras que contiene el artículo.

Cada texto se relaciona con otras entidades que indican la edición a la que pertenece, la empresa que lo ha publicado, su origen, la publicación donde ha aparecido (además del cuadernillo, sección y columna). También se relaciona con el tipo de artículo que es (entrevista, investigación, etc.).

Sección de la base de datos EMMA relacionada con las imágenes.

El segundo diagrama E-R (Fig 3.22) se refiere al almacenamiento de imágenes. El elemento principal es la entidad FOTOS que modela el almacenamiento de cada una de las fotografías en EMMA. Este diagrama está extraído del modelo general del periódico, y no se ha creado para este PFC.

Cada fotografía se relaciona con su ruta física real en el servidor donde se almacena el archivo, también con palabras clave que definen su contenido, el autor de la fotografía, y la empresa que posee sus derechos. Los atributos de la fotografía ayudarán a identificar a qué personaje pertenece, fecha en que fue tomada y en qué contexto.

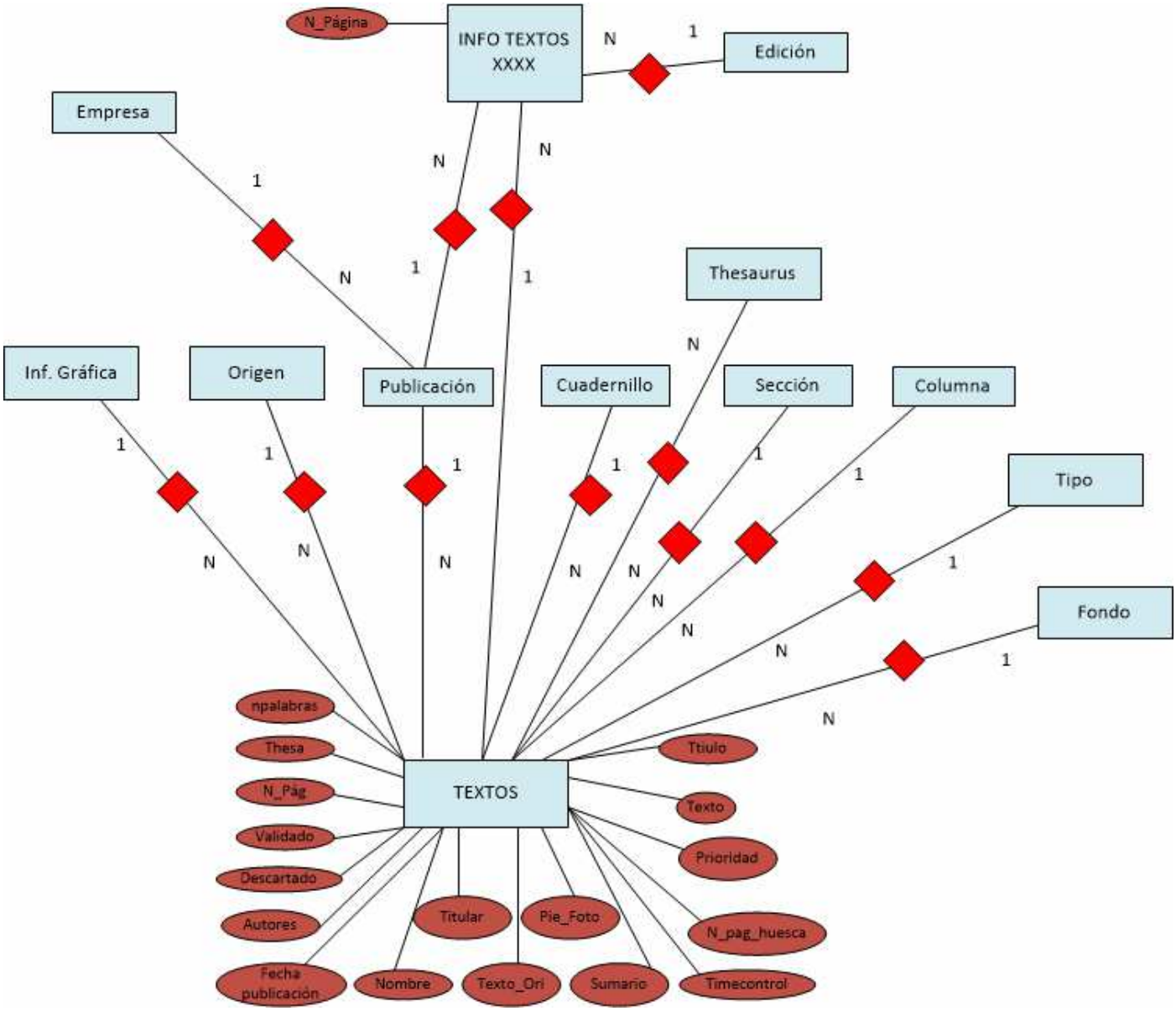

Figura 3.21 Diagrama Entidad-Relación de las entidades ya existentes en el periódico para el almacenamiento de textos en la BD EMMA.

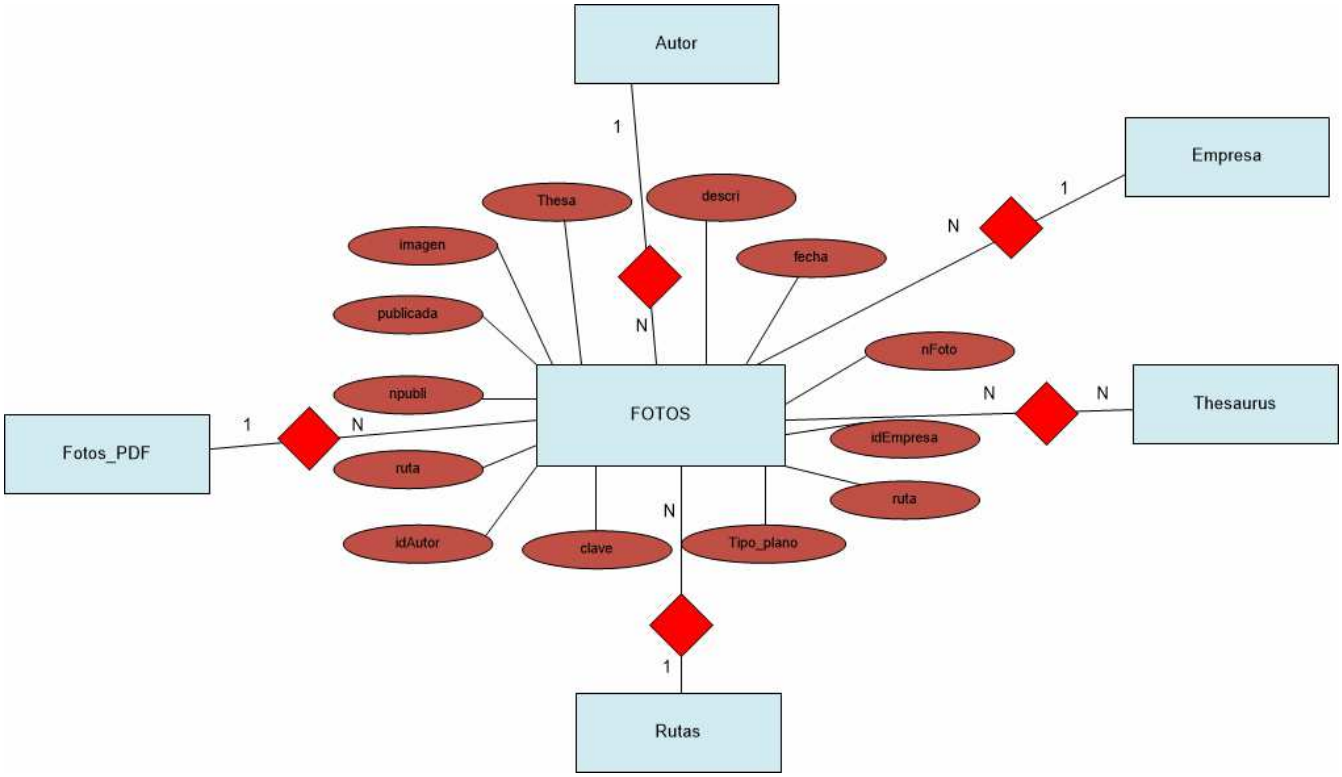

Figura 3.22 Diagrama Entidad-Relación para las fotografías en EMMA.

Tanto las entidades FOTOS como TEXTOS tienen un identificador que apunta a uno o varios elementos del tesauro. A su vez, cada elemento del tesauro tiene que corresponderse con un personaje del CATALOGO_NE (catálogo de *Named Entities*) que se ha añadido a la BD en el desarrollo de este PFC.

Esto se ha hecho porque se observa que en el tesauro los personajes pueden aparecer varias veces con el nombre escrito de diferentes formas, y también de forma diferente cuando se refieren a fotos o textos. De esta manera se garantiza que en la búsqueda dentro del archivo documental siempre se extraiga información del personaje deseado.

Restricciones de los datos:

- Un artículo pertenece siempre a una publicación, cuadernillo, sección, edición y empresa.
- Un artículo tiene información de las páginas para cada edición a la que pertenezca.
- Si un cuadernillo pertenece a X ediciones, todas ellas deben relacionarse con la misma publicación.
- La sección asociada con un artículo estará asociada a la misma publicación que el artículo.
- El cuadernillo asociado con un artículo, debe estar asociado con todas las ediciones a las que pertenezca el artículo.
- Un artículo pertenecerá a ediciones según la siguiente regla. Si pertenece a la edición general, pertenecerá a todas las demás ediciones que hayan sido lanzadas ese mismo día. Si pertenece a una edición no general, sólo pertenece a esa.

Entidades añadidas a EMMA para el funcionamiento del generador de fichas biográficas.

Durante el desarrollo de este PFC, la aportación ha sido la creación de dos nuevas entidades: un catálogo de *Named Entities* y una entidad que enlaza las noticias más relevantes con cada personaje del catálogo anterior al que se refieren.

A continuación, se puede ver el diagrama Entidad Relación (Fig. 3.23) para el catálogo de Named Entities (este catálogo se describe en la Sección 3.3.1). Esta es la primera de las dos entidades que se añaden a EMMA durante el desarrollo de este PFC:

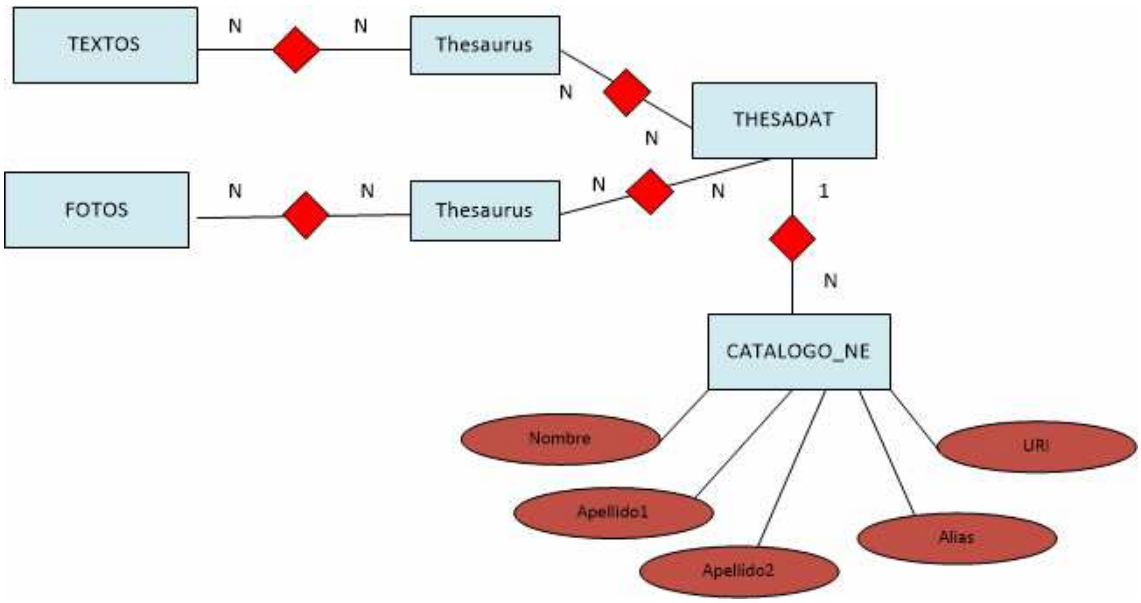

Figura 3.23 Diagrama Entidad-Relación para el catálogo de Named Entities.

En él se observa cómo las palabras clave de cada texto y fotografía, que coinciden con elementos pertenecientes al tesauro de textos y al tesauro de fotos del archivo documental, se relacionan con un listado de Named Entities almacenados en la entidad CATALOGO_NE. Así se deben poder identificar inequívocamente cada uno de los personajes que quiere localizar el usuario de la herramienta a la hora de buscar dentro del archivo documental.

La segunda entidad incorporada a EMMA es el *Ranking (Fig. 3.24)*:

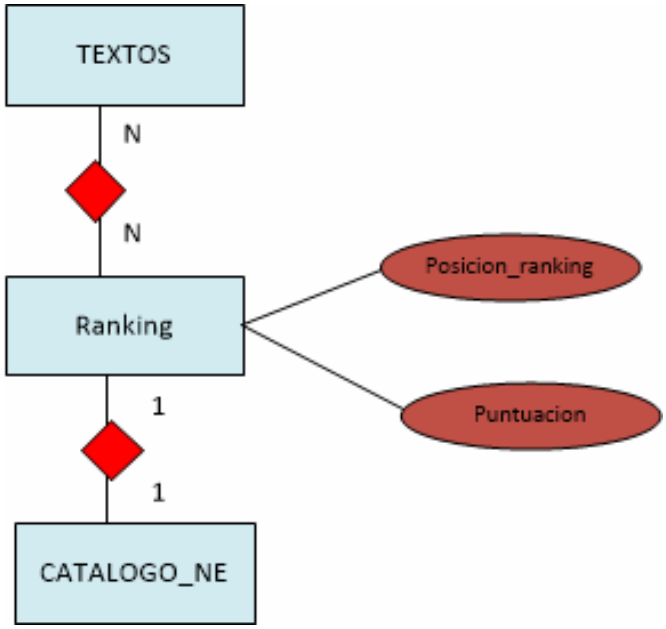

Figura 3.24 Diagrama Entidad-Relación para el Ranking de noticias relevantes.

Cada personaje del catálogo de *Named Entities* tiene una única entrada en el *Ranking*, que se relaciona con N noticias, identificadas como las más relevantes hasta ese momento para el personaje.

B2.6 Técnicas de extracción de información.

Ya se han nombrado anteriormente las técnicas estudiadas (ver Anexo A, apartado 1.3). La decisión tomada, para identificar a los personajes es la de crear un catálogo previo de entidades de nombre que nos ayudará a resolver los problemas de coherencia que nos encontramos entre el tesauro y las keywords del archivo documental. Para localizar en los textos otros datos relevantes, elegimos unas estrategias simples:

FECHA DE NACIMIENTO / LUGAR

a) Extracción de fechas

-Búsqueda de los distintos tipos de fechas: dd/mm/aaaa, dd-mm-aaaa, aaaa (año)….
-Normalmente acompañado de expresiones como “nacido/a en / el …”

b) Extracción de lugar

-Utilizando la herramienta “gazeeter”
-Precedido por “nacido en”, “oriundo de..”, ..
-Buscar un gentilicio: aragonés, turolense…(también necesitaríamos un diccionario para extraer el lugar a partir del gentilicio)

FECHA DE MUERTE / LUGAR

Igual que en punto 2 pero buscamos expresiones “fallecido el / en..”, “muerto el /en…”, “desaparecido el /en..”

Si aparece la palabra “difunto” o “desaparecido” sin preposición detrás entendemos que ha muerto pero no sabemos fecha o lugar (al menos en esa frase).

Para el lugar de la muerte como punto 2 con las expresiones mencionadas antes, y sin utilizar gentilicios.

PROFESIÓN O PUESTO QUE OCUPA

a) Identificar nombres de profesión o cargos:

-En primer lugar, veremos si podemos obtenerlo en DBpedia. Si no, buscaremos en los textos:
-Anticipando al nombre propio
-Siguiendo a expresiones como “de profesión..”, “trabajó / ha trabajado / trabaja como…”, “desempeña labores de..”, “desarrolla..”, “ocupa el cargo de…”…
-Puede aparecer sustituyendo al nombre propio: “El presidente..”
-Sería útil el uso de un diccionario de cargos o profesiones

b) Identificar fechas en los que desarrolla el cargo o profesión:

-Utilizar técnicas de puntos 2 y 3 pero que aparezcan en lugares cercanos a la profesión, y siguiendo a expresiones como: “entre (fecha) y (fecha)”.

LUGAR DE RESIDENCIA

-Utilizar técnicas de puntos 2 y 3 con las expresiones “Residente en”, “reside en..”, “actualmente vive en…”, “su lugar de residencia es…”, etc principalmente de la información de Wikipedia, antes que buscar en el archivo documental.

B3. DISEÑO

En este capítulo se descomponen, organizan y describen desde diferentes perspectivas los elementos que formarán la herramienta a partir del análisis hecho en el capítulo anterior.

B3.1 Diagrama de componentes.

Finalmente, se ha tomado la decisión de construir una aplicación que en primer lugar será un ejecutable que se instale en los puestos de los usuarios y que más adelante, se incorpore a la plataforma EMMA como un servicio web más. La estructura podemos verla en el diagrama de componentes:

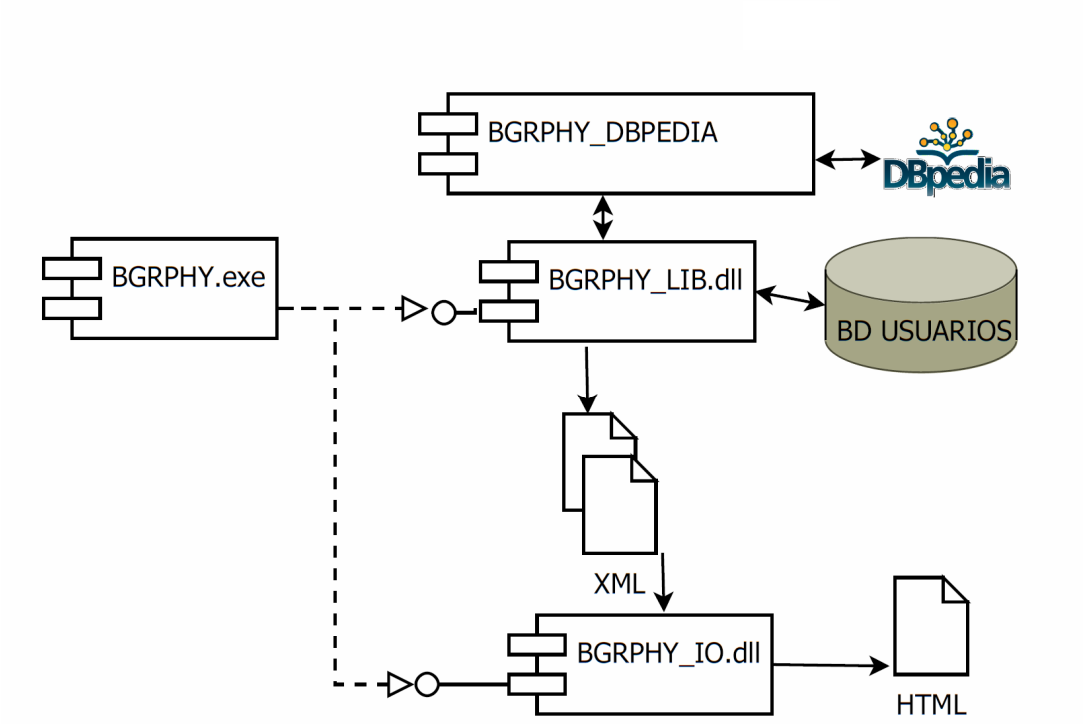

Figura 3.24 Diagrama de componentes.

Este ejecutable llama a la librería BGRPHY_LIB escrita en .NET que hará las búsquedas en el archivo documental, y también comprobará la autorización de los usuarios técnicos de informática que quieran modificar la configuración de la aplicación. Esta librería se apoyará para las búsquedas en DBpedia en otra librería (BGRPHY_DBPEDIA). Cuando haya recopilado toda la información, la librería BGRPHY_IO le dará formato, recuperándola de un archivo XML y transformándolo según los parámetros de configuración (extensión, tipo de usuario final) en un HTML.

B3.2 Diseño de procesos.

Se presentan aquí los procesos de más bajo nivel en los casos de uso anteriores. Se muestran por orden de uso en la aplicación de generación de fichas biográficas. En cada diagrama aparecerán recuadrados en rojo los procesos que se describirán a continuación.

Listado de los procesos que se explican más adelante:

PROCESOS

Caso uso 1.1 (nivel 3)
- 1.1.1. Lanzar formulario de recogida de datos
- 1.1.2. Formatear datos
- 1.1.3. Visor de pantalla final

Caso uso 1.2 (nivel 3)
- 1.2.1. Lanzar formulario de recogida de datos
- 1.2.2. Formatear datos
- 1.2.3. Visor de pantalla final

Caso uso 1 (nivel 2)
- 1.3. Generador XML
- 1.4. Inicializador de interfaz
- 1.5. Procesador XML

Caso uso 2.1 (nivel 3)
- 2.1.1. Interfaz de entrada
- 2.1.2. Comprobar datos

Caso uso 2.2 (nivel 3)
- 2.2.1. Búsqueda en tesauro
- 2.2.2. Búsqueda en catalogo de Named Entities
- 2.2.3. Búsqueda en DBpedia español
- 2.2.4. Búsqueda en DBpedia ingles

Caso uso 2 (nivel 2)
- 2.3. Desambiguador

Caso uso 3.1 (nivel 3)
- 3.1.1. Extraer noticias recientes
- 3.1.2. Extraer noticias relevantes
- 3.1.3. Extraer fotografías recientes
- 3.1.4. Extraer fotografías primer plano

Caso uso 3.2 (nivel 3)
- 3.2.1. Recuperación de información en wikipedia
- 3.2.2. Recuperación de imágenes
- 3.2.3. Recuperación de vídeos
- 3.2.4. Recuperación de libros
- 3.2.5. Recuperación de blogs
- 3.2.6. Recuperación de redes sociales
- 3.2.7. Recuperación de web personal

Caso uso 3.3 (nivel 3)
- 3.3.1. Búsqueda en DBpedia español
- 3.3.2. Búsqueda en DBpedia inglés

Caso uso 3 (nivel 2)
- 3.4. Generador de XML

Caso uso nivel 1
- 4. Configurador

DISEÑO PARA CASO DE USO NIVEL 3: 1.1 HERRAMIENTA PRESENTACIÓN PERIODISTAS

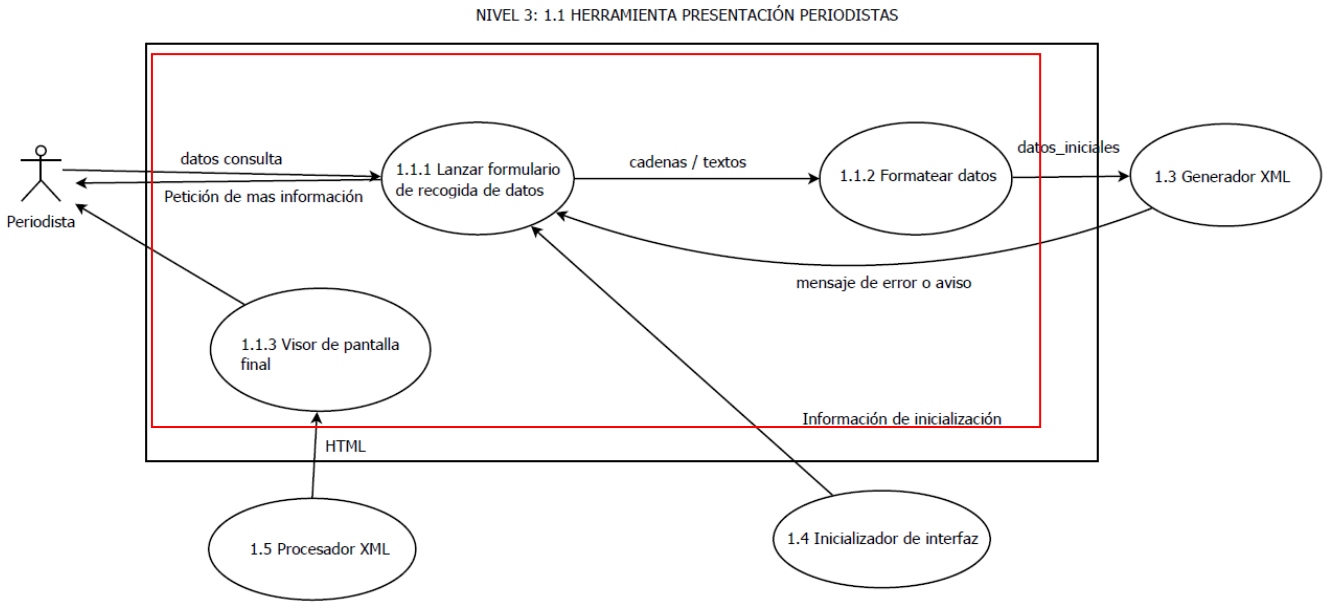

Figura 3.25 Diagrama nivel 3, 1.1.

1.1.1. LANZAR FORMULARIO DE RECOGIDA DE DATOS

Descripción del proceso:	
Muestra en pantalla un formulario que nos permite recoger del usuario los datos necesarios sobre el personaje del que se realizará la ficha biográfica. Presenta el mensaje de solicitud de más información si el proceso 1.3 Generador XML nos indica que los datos son incompletos.	
Entradas del sistema	**Formato**
Nombre_personaje	Texto
Salidas del sistema	**Formato**
Petición_más_información	Texto
Nombre	Texto
Primer apellido	Texto
Segundo apellido	Texto
Alias	Texto

1.1.2. FORMATEAR DATOS

Descripción del proceso:	
Recibe los textos del formulario de recogida de datos y los transforma en cadenas de mayúsculas y sin acentos.	
Entradas del sistema	**Formato**
Nombre_personaje	Texto
Salidas del sistema	**Formato**
Nombre	Texto
Primer apellido	Texto
Segundo apellido	Texto
Alias	Texto

1.1.3. VISOR DE PANTALLA FINAL

Descripción del proceso:	
Muestra en pantalla el archivo HTML con la ficha biográfica resultante de todo el proceso.	
Entradas del sistema	**Formato**
Ficha biográfica	HTML
Salidas del sistema	**Formato**
Ficha biográfica	Web

CASO DE USO NIVEL 3: 1.2 HERRAMIENTA PRESENTACIÓN DOCUMENTALISTAS

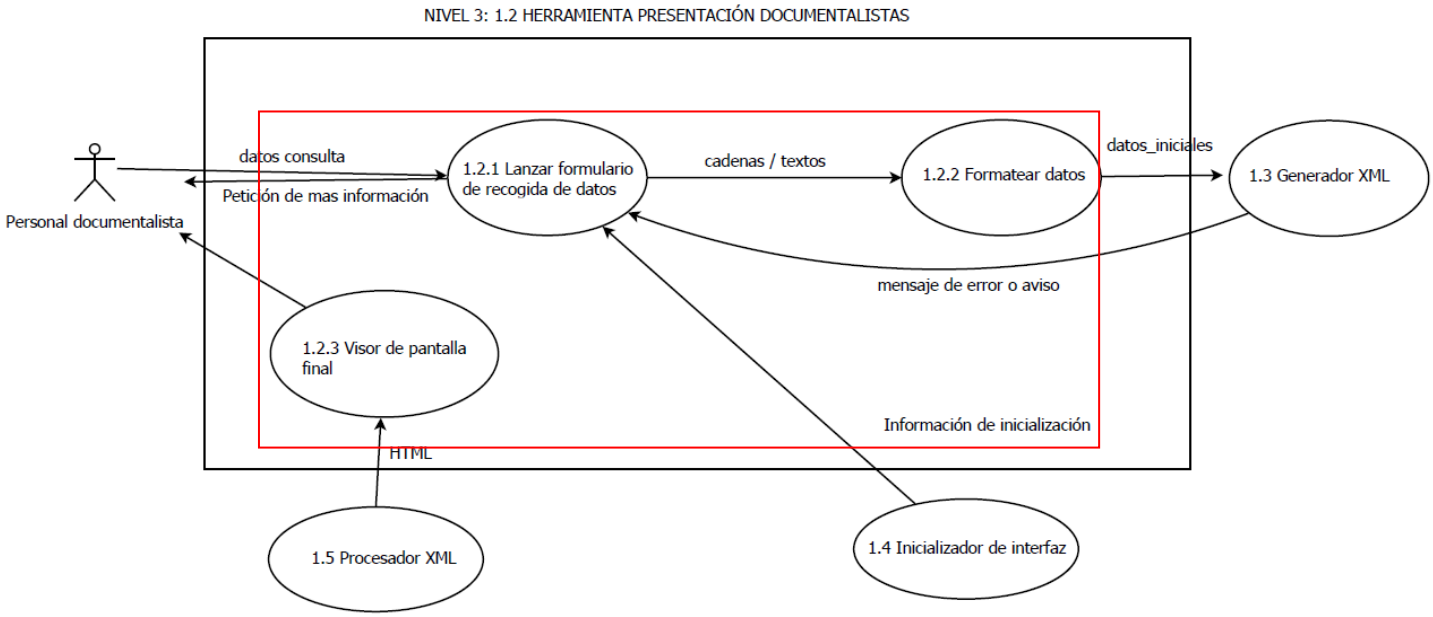

Figura 3.26 Diagrama nivel 3, 1.2.

1.2.1. LANZAR FORMULARIO DE RECOGIDA DE DATOS

Descripción del proceso:	
Muestra en pantalla un formulario que nos permite recoger del usuario los datos necesarios sobre el personaje del que se realizará la ficha biográfica. Presenta el mensaje de solicitud de más información si el proceso 1.3 Generador XML nos indica que los datos son incompletos.	
Entradas del sistema	**Formato**
Nombre	Texto
Primer apellido	Texto
Segundo apellido	Texto
Alias	Texto
Fecha_inicio_busqueda	Texto
Fecha_fin_busqueda	Texto
Salidas del sistema	**Formato**
Petición_más_información	Texto
Nombre	Texto
Primer apellido	Texto
Segundo apellido	Texto
Alias	Texto

1.2.2. FORMATEAR DATOS

Descripción del proceso:	
Recibe los textos del formulario de recogida de datos y los transforma en cadenas de mayúsculas y sin acentos.	
Entradas del sistema	**Formato**
Nombre	Texto
Primer apellido	Texto
Segundo apellido	Texto
Alias	Texto
Salidas del sistema	**Formato**
Nombre	Texto
Primer apellido	Texto
Segundo apellido	Texto
Alias	Texto

1.2.3. VISOR DE PANTALLA FINAL

Descripción del proceso:	
Muestra en pantalla el archivo HTML con la ficha biográfica resultante de todo el proceso.	
Entradas del sistema	**Formato**
Ficha biográfica	HTML
Salidas del sistema	**Formato**
Ficha biográfica	Web

CASO DE USO NIVEL 2: 1 INTERFAZ DE USUARIO

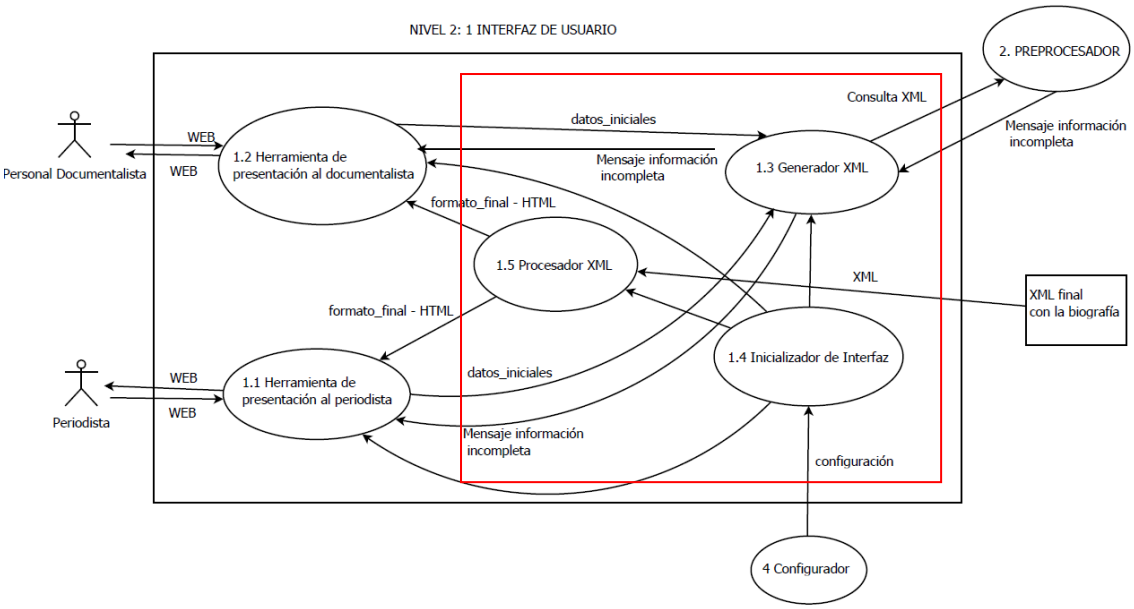

Figura 3.27 Diagrama nivel 2, 1.

1.3. GENERADOR DE XML

Descripción del proceso:	
Transforma las cadenas de datos sobre el personaje en un archivo XML.	
Entradas del sistema	**Formato**
Nombre	Texto
Primer apellido	Texto
Segundo apellido	Texto
Alias	Texto
Configuración	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.ruta_archivos_xml	Entero
Configuración.prefijo_nombre_xml_datos_iniciales	Entero
Configuración.contador_instancia	Entero
Salidas del sistema	**Formato**
Datos_iniciales	Archivo XML

Estructura del fichero de salida Datos_iniciales en formato XML:

<XML>
<datos_iniciales>
<Nombre> Nombre del personaje que buscamos **</Nombre>**
<Apellido1> Primer apellido del personaje **</Apellido1>**
<Apellido2> Segundo apellido del personaje (opcional) **</Apellido2 >**
<Alias> Alias del personaje (opcional) **</Alias>**
</datos_iniciales>
</XML>

1.4. INICIALIZADOR DE INTERFAZ

Descripción del proceso:	
Lee el archivo XML con la configuración introducida por el personal de informática, y la carga en memoria mediante una instancia al objeto configuración.	
Entradas del sistema	**Formato**
Configuración	Archivo XML
Salidas del sistema	**Formato**
configuración	Objeto configuracion

1.5. PROCESADOR DE XML

Descripción del proceso:	
Lee el archivo XML con la ficha biográfica de resultados, y la transforma, según los parámetros de configuración, en un archivo HTML.	
Entradas del sistema	**Formato**
Ficha_personaje	Archivo XML
Configuración	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.ruta_archivos_xml	Entero
Configuración.prefijo_nombre_xml_resultado	Entero
Configuración.contador_instancia	Entero
Salidas del sistema	**Formato**
Ficha_personaje	Archivo HTML

Estructura del fichero de salida ficha_personaje en formato XML:

<XML>
<**ficha_personaje**>

<**DatosBiograficos**>

<**NombreCompleto**>
Nombre y apellidos del personaje que buscamos
<**Nombre**> Nombre de pila (opcional) </**Nombre**>
<**Apellido1**> Primer apellido </**Apellido1**>
<**Apellido2**>Segundo apellido (opcional)</**Apellido2**>
<**Alias**>Alias o sobrenombre del personaje (opcional)</**Alias**>
</**NombreCompleto**>

<**FechaNacimiento**>
<**Dia**> Día de nacimiento del personaje </**Dia**>
<**Mes**> Mes en el que nace el personaje </**Mes**>
<**Año**> Año en el que nace el personaje </**Año**>
</**FechaNacimiento**>

<**FechaMuerte**>
(si es el caso)
<**Dia**> Día de la muerte del personaje </**Dia**>

<**Mes**> Mes en el que muere el personaje </**Mes**>
<**Año**> Año en el que muere el personaje </**Año**>
</**FechaMuerte**>

<**LugarNacimiento**>
<**Poblacion**> Ciudad de nacimiento </**Poblacion**>
<**Pais**> País de nacimiento </**Pais**>
</**LugarNacimiento**>

<**LugarMuerte**>
(si ha muerto)
<**Poblacion**> Ciudad donde muere </**Poblacion**>
<**Pais**> País donde muere </**Pais**>
</**LugarMuerte**>

<**EstadoCivil**> Estado civil </**EstadoCivil**>

<**Cargos**>
<**Cargo1**>
<**Puesto**> Nombre del cargo </**Puesto**>
<**Organizacion**>Organización o empresa donde desempeña el cargo **</Organizacion>**
<**Fechas**>
<**FechaIni**>Fecha inicio en el cargo</**FechaIni**>
<**FechaFin**> Fecha final en el cargo <**FechaFin**>
</**Fechas**>
<**PredecesorCargo**>Predecesor en el cargo</**PredecesorCargo**>
<**SucesorCargo**>Sucesor en el cargo</**SucesorCargo**>
</**Cargo1**>
…
<**CargoN**>
</**CargoN**>
<**Cargos**>

</**DatosBiograficos**>

<**ImagenesInternet**>
<**Imagen1**>
<**Fecha**>Fecha de la imagen</**Fecha**>
<**Pie**>Texto del pie de foto</**Pie**>
<**Autor**>Autor de la foto</**Autor**>
<**Fichero**>URL del fichero</**Fichero**>
</**Imagen1**>
….
<**ImagenN**>
</**ImagenN**>
</**ImagenesInternet**>
<**ImagenesArchivo**>
<**FotosPrimerPlano**>
<**Imagen1**>
<**id**>Identificador de la imagen</**id**>
<**Fecha**>Fecha en que se toma la imagen</**Fecha**>
<**Pie**>Texto del pie de foto </**Pie**>
<**Autor**>Autor de la foto</**Autor**>
<**Fichero**>Nombre del fichero (jpg) que contiene la foto</**Fichero**>
<**Publicada**>Nos indica si está o no publicada</**Publicada**>
</**Imagen1**>

```
….
<ImagenN>
</ImagenN>
</FotosPrimerPlano>
<FotosRelacionadas>
<Imagen1>
<id>Identificador de la imagen</id>
<Fecha>Fecha en que se toma la imagen</Fecha>
<Pie>Texto del pie de foto </Pie>
<Autor>Autor de la foto</Autor>
<Fichero>Nombre del fichero (jpg) que contiene la foto</Fichero>
<Publicada>Nos indica si está o no publicada</Publicada>
</Imagen1>
….
<ImagenN>
</FotosRelacionadas>
<FotosRecientes>
<Imagen1>
<id>Identificador de la imagen</id>
<Fecha>Fecha en que se toma la imagen</Fecha>
<Pie>Texto del pie de foto </Pie>
<Autor>Autor de la foto</Autor>
<Fichero>Nombre del fichero (jpg) que contiene la foto</Fichero>
<Publicada>Nos indica si está o no publicada</Publicada>
</Imagen1>
….
<ImagenN>
</FotosRecientes>
</ImagenesArchivo>
<Noticias>
<NoticiasRecientes>
<Noticia1>
<id>Identificador de la noticia</id>
<Fecha>Fecha en que se publica la noticia</Fecha>
<Titulo>Titular de la noticia</Titulo>
<Subtitulo>Segundo titular de la noticia</Subtitulo>
<Texto>Texto de la noticia</Texto>
<Resumen>Resumen del texto de la noticia</Resumen>
<Autor>Autor de la noticia</Autor>
</Noticia1>
…
<NoticiaN>
</NoticiaN>
</NoticiasRecientes>

<NoticiasRelevantes>
<Noticia1>
<id>Identificador de la noticia</id>
<Fecha>Fecha en que se publica la noticia</Fecha>
<Titulo>Titular de la noticia</Titulo>
<Subtitulo>Segundo titular de la noticia</Subtitulo>
<Texto>Texto de la noticia</Texto>
<Resumen>Resumen del texto de la noticia</Resumen>
<Autor>Autor de la noticia</Autor>
</Noticia1>
…
<NoticiaN>
</NoticiaN>
```

```
        </NoticiasRelevantes>

        <Entrevistas>
        <Noticia1>
        <id>Identificador de la noticia</id>
        <Fecha>Fecha en que se publica la noticia</Fecha>
        <Titulo>Titular de la noticia</Titulo>
        <Texto>Texto de la noticia</Texto>
        <Resumen>Resumen del texto de la noticia</Resumen>
        <Autor>Autor de la noticia</Autor>
        </Noticia1>
        …
        <NoticiaN>
        </NoticiaN>
        </Entrevistas>
</Noticias>

<Paginas>
        <Pagina1>
        <id>Identificador de la página</id>
        <Fecha>Fecha de publicación</Fecha>
        <NumPag>Número de página</NumPag>
        <Fichero>Nombre del archive .pdf que contiene la imagen de la
        página</Fichero>
        </Pagina1>
        <PaginaN>
        </PaginaN>
</Paginas>

<Internet>
        <Wikipedia>
                <URL_página>
                URL de la página del personaje en wikipedia
                </URL_página>
                <Texto>
                Texto extraído de wikipedia que queremos incorporar a la ficha
                biográfica
                </Texto>
        </Wikipedia>

        <Videos>
                <Video1>
                        <Nombre_archivo>Nombre del archivo donde
                        está el vídeo</Nombre_archivo>
                        <URL_video>URL donde se encuentra el vídeo en
                Internet</ URL_vídeo>
                        <Ruta_almacenamiento_video> Carpeta donde se
                guarda el archivo de vídeo
                        </Ruta_almacenamiento_video>
                </Video1>
                …
        </Videos>

        <Libros>
                <Libro1>
                        <URL_libro>URL donde se encuentra el libro en
                        Internet </URL_libro>
                         <Titulo> Título del libro </Titulo>
```

```
                    </Libro1>
                    …
                </Libros>

                <Blogs>
                    <Blog1>
                        <URL_blog>URL donde se encuentra el blog en Internet </URL_blog>
                        <Titulo> Título del blog </Titulo>
                    </Blog1>
                    …
                </Blogs>

                <RedesSociales>
                    <URL_Facebook> URL donde encontrar al personaje en Facebook</URL_Facebook>
                    <URL_Twitter> URL donde encontrar al personaje en Twitter </URL_Twitter>
                    <URL_Linkedin> URL donde encontrar al personaje en Linkedin </URL_Linkedin>
                    <URL_Googleplus> URL donde encontrar al personaje en Google+ </URL_Googleplus>
                </RedesSociales>

                <WebPersonal>
                <URL>URL de la web personal que aparezca primero en la búsqueda en Google </URL>
                </WebPersonal>

            </Internet>
        </ficha_personaje>
</XML>
```

CASO DE USO NIVEL 3: 2.1 PROCESAR XML

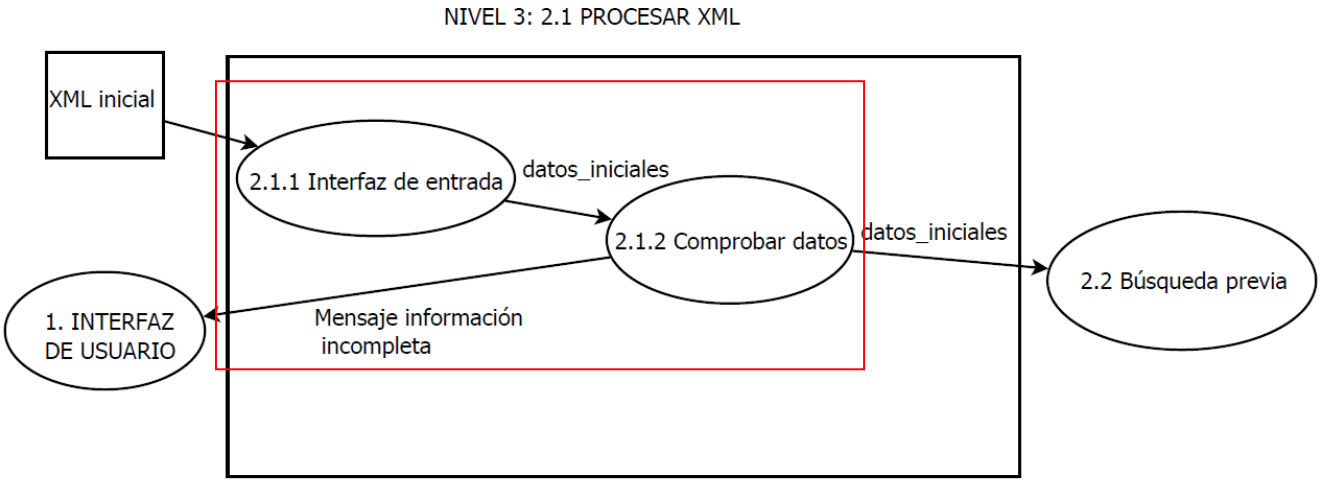

Figura 3.28 Diagrama nivel 3, 2.1.

2.1.1. INTERFAZ DE ENTRADA

Descripción del proceso:	
Lee el archivo XML datos_iniciales con los datos introducidos por el usuario.	
Entradas del sistema	**Formato**
Datos_iniciales	Archivo XML
Salidas del sistema	**Formato**
Datos_iniciales	Objeto datos_iniciales

2.1.2. COMPROBAR DATOS

Descripción del proceso:	
Comprueba si en los datos está como mínimo, el primer apellido para realizar una búsqueda.	
Entradas del sistema	**Formato**
Datos_iniciales	Objeto datos_iniciales
Atributos del objeto que utiliza el proceso:	
Datos_iniciales.Nombre	Texto
Datos_iniciales.Apellido1	Texto
Datos_iniciales.Apellido2	Texto
Datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Datos_iniciales	Objeto datos_iniciales
Mensaje información incompleta	Texto

CASO DE USO NIVEL 3: 2.2 BUSQUEDA PREVIA

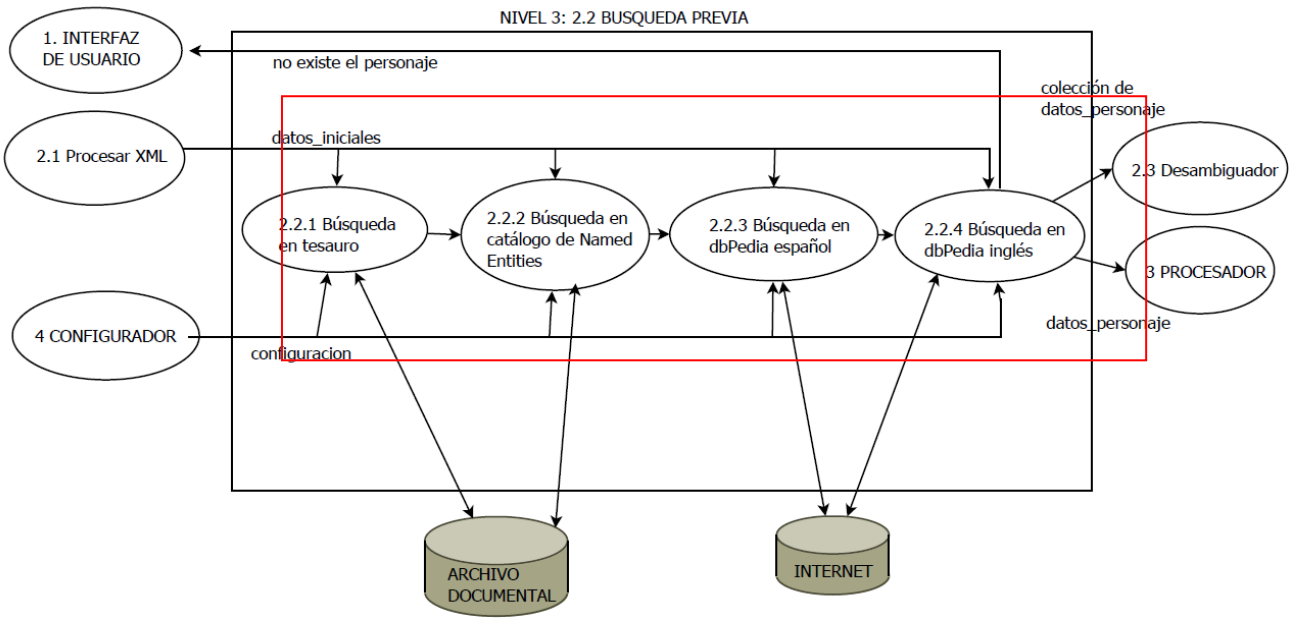

Figura 3.29 Diagrama nivel 3, 2.2.

2.2.1. BÚSQUEDA EN TESAURO

Descripción del proceso:	
Con los datos del personaje en el objeto datos_iniciales, busca apariciones de dicho personaje en el tesauro de la base de datos. Almacena y cuenta cada aparición del personaje y guarda su fecha más reciente.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.fecha_inicio_busqueda_noticias	Fecha
Configuración.fecha_fin_busqueda_noticias	Fecha
Configuración.gestor_BD	Objeto gestor_BD
Configuración.Prefijo_Tablas_tesauro	Texto
Configuración.Prefijo_Tablas_textos	Texto
Datos_iniciales	Objeto datos_iniciales
Atributos del objeto que utiliza el proceso:	
Datos_iniciales.Nombre	Texto
Datos_iniciales.Apellido1	Texto
Datos_iniciales.Apellido2	Texto
Datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Datos_personaje	Objeto datos_personaje

2.2.2. BÚSQUEDA EN CATALOGO DE NAMED ENTITIES

Descripción del proceso:	
Con los datos del personaje en el objeto datos_iniciales, busca apariciones de dicho personaje en el catálogo de Named Entities de la base de datos. Almacena y cuenta cada aparición del personaje y guarda su fecha más reciente.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.gestor_BD	Objeto gestor_BD
Configuración.Tabla_catalogo_NE	Texto
Datos_iniciales	Objeto datos_iniciales
Atributos del objeto que utiliza el proceso:	
Datos_iniciales.Nombre	Texto
Datos_iniciales.Apellido1	Texto
Datos_iniciales.Apellido2	Texto
Datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Datos_personaje	Objeto datos_personaje

2.3.3. BÚSQUEDA EN DBPEDIA ESPAÑOL

Descripción del proceso:	
Con los datos del personaje en el objeto datos_iniciales, busca apariciones de dicho personaje en el sitio DBpedia en Español. Almacena y cuenta cada aparición del personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.busqueda_DBpedia_español	Texto
Datos_iniciales	Objeto datos_iniciales
Atributos del objeto que utiliza el proceso:	
Datos_iniciales.Nombre	Texto
Datos_iniciales.Apellido1	Texto
Datos_iniciales.Apellido2	Texto
Datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Datos_personaje	Objeto datos_personaje

2.3.4. BÚSQUEDA EN DBPEDIA INGLÉS

Descripción del proceso:	
Con los datos del personaje en el objeto datos_iniciales, busca apariciones de dicho personaje en el sitio DBpedia en Inglés. Almacena y cuenta cada aparición del personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.busqueda_DBpedia_ingles	Texto
Datos_iniciales	Objeto datos_iniciales
Atributos del objeto que utiliza el proceso:	
Datos_iniciales.Nombre	Texto
Datos_iniciales.Apellido1	Texto
Datos_iniciales.Apellido2	Texto
Datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Datos_personaje	Objeto datos_personaje

CASO DE USO NIVEL 2: 2 PREPROCESADOR

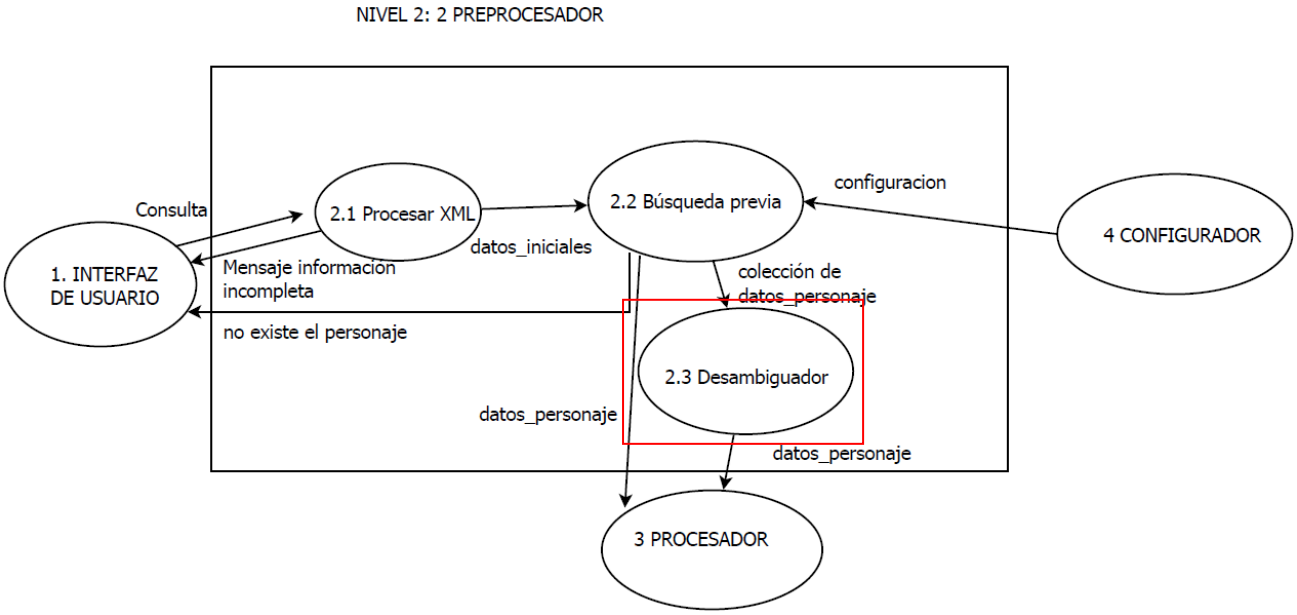

Figura 3.30 Diagrama nivel 2, 2.

2.3 DESAMBIGUADOR

Descripción del proceso:	
Si recibe un solo objeto datos_personaje, lo devuelve tal cual. Si recibe una colección, escoge el más frecuente en apariciones, incluyendo la aparición en DBpedia español e inglés. En caso de empate, escoge el más reciente en fechas de aparición.	
Entradas del sistema	**Formato**
Datos_personaje	Objeto datos_personaje
Colección datos_personaje	Colección de objetos datos_personaje
Salidas del sistema	**Formato**
Datos_personaje	Objeto datos_personaje

CASO DE USO NIVEL 3: 3.1 BUSQUEDA EN ARCHIVO DOCUMENTAL

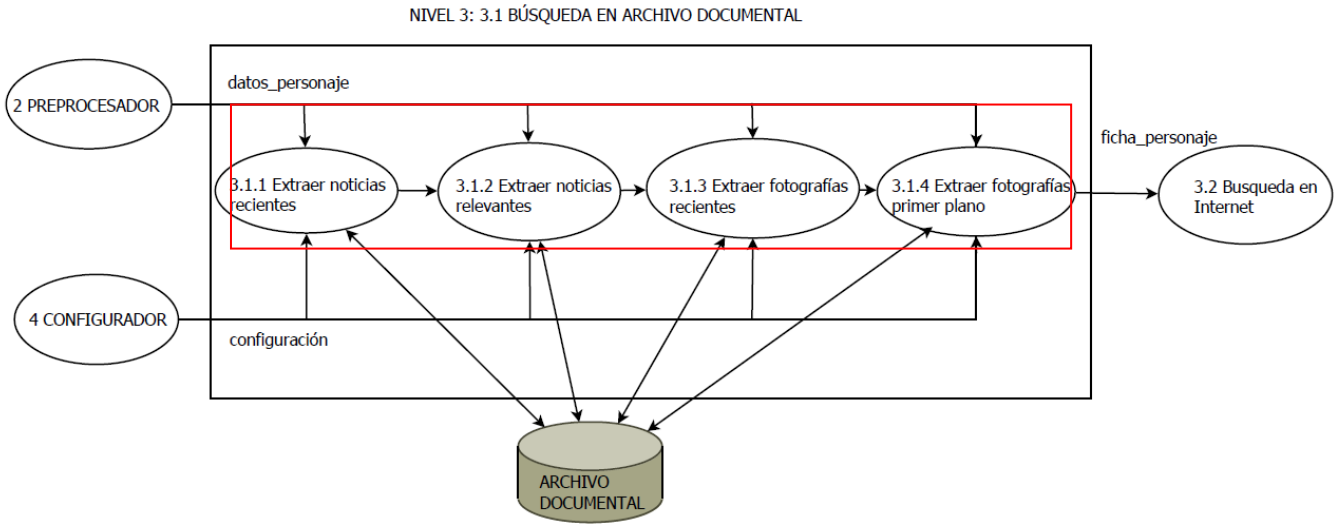

Figura 3.31 Diagrama nivel 3, 3.1.

3.1.1. EXTRAER NOTICIAS RECIENTES

Descripción del proceso:	
Lee los datos del personaje y extrae de la base de datos EMMA, según los límites de fechas y el número máximo de noticias que marca el configurador, las noticias más recientes que nombran al personaje buscado. Almacena los resultados en el objeto ficha_personaje.	
Recursos:	
Archivo documental (EMMA)	Base de Datos
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.fecha_inicio_busqueda_noticias	Fecha
Configuración.fecha_fin_busqueda_noticias	Fecha
Configuración.gestor_BD	Objeto gestor_BD
Configuración.Prefijo_Tablas_textos	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.1.2. EXTRAER NOTICIAS RELEVANTES

Descripción del proceso:	
Lee los datos del personaje y extrae de la base de datos EMMA, según los límites de fechas y el número máximo de noticias que marca el configurador, las noticias más relevantes que nombran al personaje buscado. En el objeto configuración guardamos unos listados de pesos según unos umbrales para cada uno de los parámetros que valoran la relevancia de una noticia. Ésta se pondera en función de las apariciones del personaje en el resumen, en el texto de la noticia, en el titular y en el pie de foto. También se valora el número de palabras de la noticia, la sección y la página donde aparece. Almacena los resultados en el objeto ficha_personaje.	
Recursos:	
Archivo documental (EMMA)	Base de Datos
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.fecha_inicio_busqueda_noticias	Fecha
Configuración.fecha_fin_busqueda_noticias	Fecha
Configuración.gestor_BD	Objeto gestor_BD
Configuración.Prefijo_Tablas_textos	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.1.3. EXTRAER FOTOGRAFÍAS RECIENTES

Descripción del proceso:	
Lee los datos del personaje y extrae de la base de datos EMMA, según los límites de fechas y el número máximo de fotografías que marca el configurador, las fotografías más recientes que muestran al personaje buscado. Almacena los resultados en el objeto ficha_personaje.	
Recursos:	
Archivo documental (EMMA)	Base de Datos
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.fecha_inicio_busqueda_fotos	Fecha
Configuración.fecha_fin_busqueda_fotos	Fecha
Configuración.gestor_BD	Objeto gestor_BD
Configuración.Prefijo_Tablas_fotos	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**

Ficha_personaje	Objeto ficha_personaje

3.1.4. EXTRAER FOTOGRAFÍAS PRIMER PLANO

Descripción del proceso:	
Lee los datos del personaje y extrae de la base de datos EMMA, según los límites de fechas y el número máximo de fotografías que marca el configurador, las fotografías de primer plano que muestran al personaje buscado. Almacena los resultados en el objeto ficha_personaje.	
Recursos:	
Archivo documental (EMMA)	Base de Datos
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.fecha_inicio_busqueda_fotos	Fecha
Configuración.fecha_fin_busqueda_fotos	Fecha
Configuración.gestor_BD	Objeto gestor_BD
Configuración.Prefijo_Tablas_fotos	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

CASO DE USO NIVEL 3: 3.2 BUSQUEDA EN INTERNET

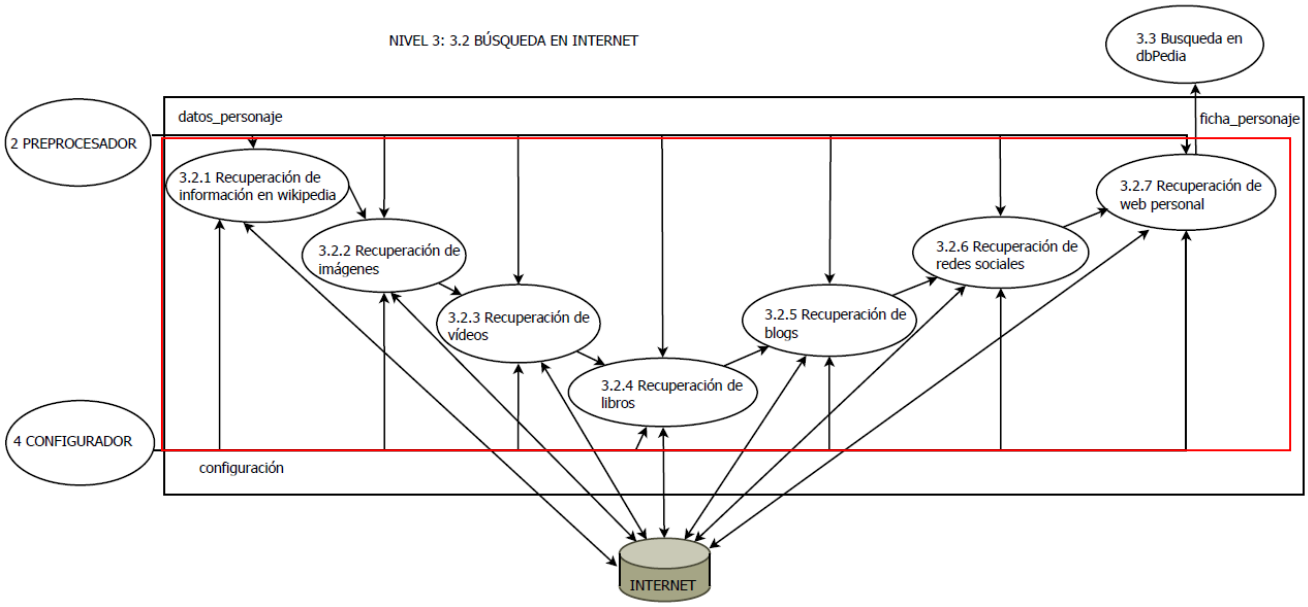

Figura 3.32 Diagrama nivel 3, 3.2.

3.2.1. RECUPERACIÓN DE INFORMACIÓN EN WIKIPEDIA

Descripción del proceso:	
Lee los datos del personaje, y lo localiza en wikipedia. De la página encontrada, según los parámetros del configurador, filtra los datos y los almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_wikipedia	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.2.2. RECUPERACIÓN DE IMÁGENES

Descripción del proceso:	
Lee los datos del personaje, y lo localiza en un buscador de imágenes en Internet. Según los parámetros del configurador, filtra las imágenes y las almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_imagenes	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.2.3. RECUPERACIÓN DE VÍDEOS

Descripción del proceso:	
Lee los datos del personaje, y lo localiza en un buscador de vídeos en Internet. Según los parámetros del configurador, selecciona los vídeos y los almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_videos	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.2.4. RECUPERACIÓN DE LIBROS

Descripción del proceso:	
Lee los datos del personaje, y lo localiza en un buscador de vídeos en Internet. Según los parámetros del configurador, selecciona los vídeos y los almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_libros	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.2.5. RECUPERACIÓN DE BLOGS

Descripción del proceso:	
Lee los datos del personaje, y lo localiza en un buscador de blogs en Internet. Según los parámetros del configurador, selecciona los blogs y los almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_blogs	Texto
Datos_personaje	Objeto datos_personaje

Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.2.6. RECUPERACIÓN DE REDES SOCIALES

Descripción del proceso:	
Lee los datos del personaje, y lo localiza en un buscador de blogs en Internet. Según los parámetros del configurador, selecciona los blogs y los almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_facebook	Texto
Configuración.cadena_busqueda_twitter	Texto
Configuración.cadena_busqueda_linkedin	Texto
Configuración.cadena_busqueda_google+	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.2.7. RECUPERACIÓN DE WEB PERSONAL

Descripción del proceso:	
Lee los datos del personaje, y localiza páginas web del personaje en Internet. Según los parámetros del configurador, selecciona las páginas y las almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_web	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

CASO DE USO NIVEL 3: 3.3 BUSQUEDA EN DBPEDIA

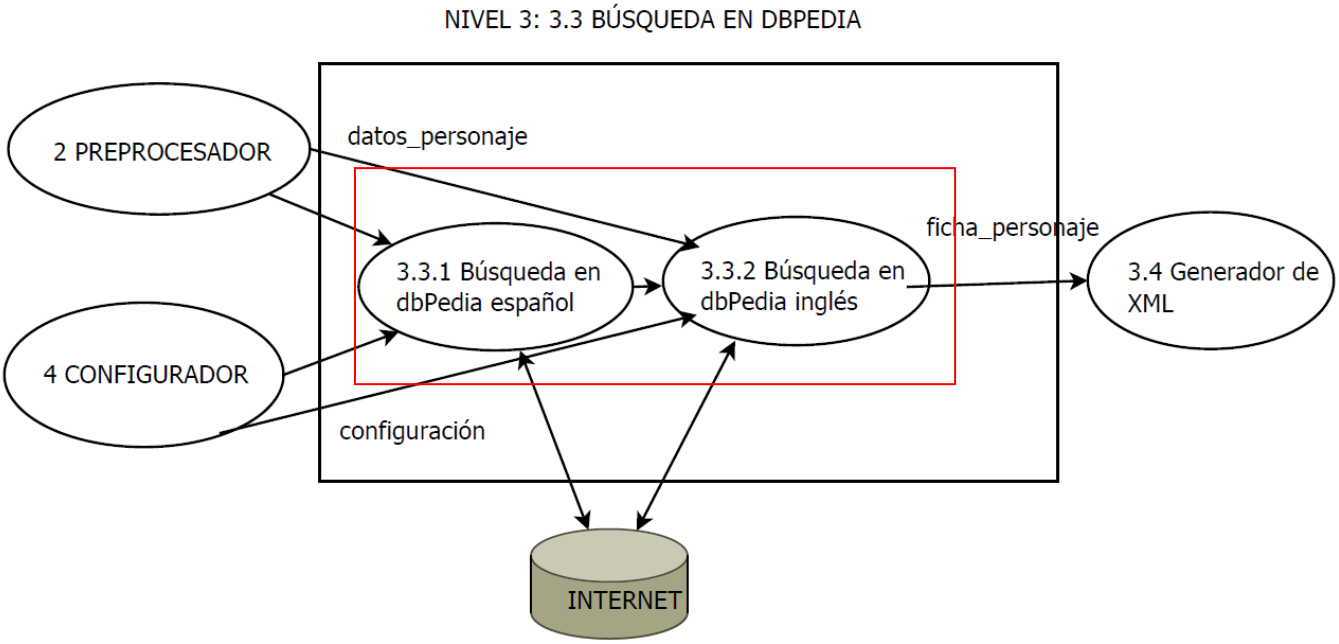

Figura 3.33 Diagrama nivel 3, 3.3.

3.3.1. BÚSQUEDA EN DBPEDIA ESPAÑOL

Descripción del proceso:	
Lee los datos del personaje, y lo busca en DBpedia en español. De la página encontrada, según los parámetros del configurador, filtra los datos y los almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_DBpedia_español	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

3.3.2. BÚSQUEDA EN DBPEDIA INGLÉS

Descripción del proceso:	
Lee los datos del personaje, y lo busca en DBpedia en inglés. De la página encontrada, según los parámetros del configurador, filtra los datos y los almacena en el objeto ficha_personaje.	
Entradas del sistema	**Formato**
configuracion	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.cadena_busqueda_DBpedia_español	Texto
Datos_personaje	Objeto datos_personaje
Atributos del objeto que utiliza el proceso:	
Datos_personaje.datos_iniciales.Nombre	Texto
Datos_personaje.datos_iniciales.Apellido1	Texto
Datos_personaje.datos_iniciales.Apellido2	Texto
Datos_personaje.datos_iniciales.Alias	Texto
Salidas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje

CASO DE USO NIVEL 2: 3 PROCESADOR

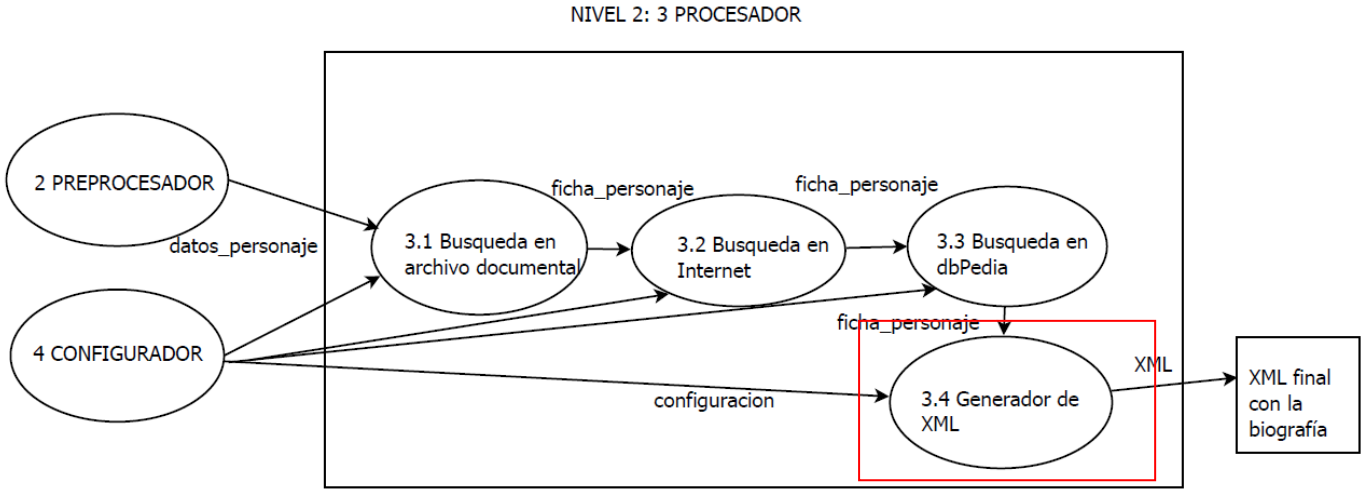

Figura 3.34 Diagrama nivel 2, 3.

3.4. GENERADOR DE XML

Descripción del proceso:	
Escribe el archivo XML con todos los datos que componen la ficha biográfica, utilizando los parámetros de configuración.	
Entradas del sistema	**Formato**
Ficha_personaje	Objeto ficha_personaje
Configuración	Objeto configuracion
Atributos del objeto que utiliza el proceso:	
Configuración.num_noticias_recientes	Entero
Configuración.num_noticias_relevantes	Entero
Configuración.num_fotos_relacionadas	Entero
Configuración.num_fotos_primer_plano	Entero
Configuración.contador_instancia	Entero
Configuración.num_blogs	Entero
Configuración.num_libros	Entero
Configuración.num_videos	Entero
Configuración.num_imagenes_internet	Entero
Configuración.ruta_archivos_xml	Entero
Configuración.prefijo_nombre_xml_resultado	Entero
Configuración.contador_instancia	Entero
Salidas del sistema	**Formato**
Ficha personaje	Archivo XML

CASO DE USO NIVEL 1:

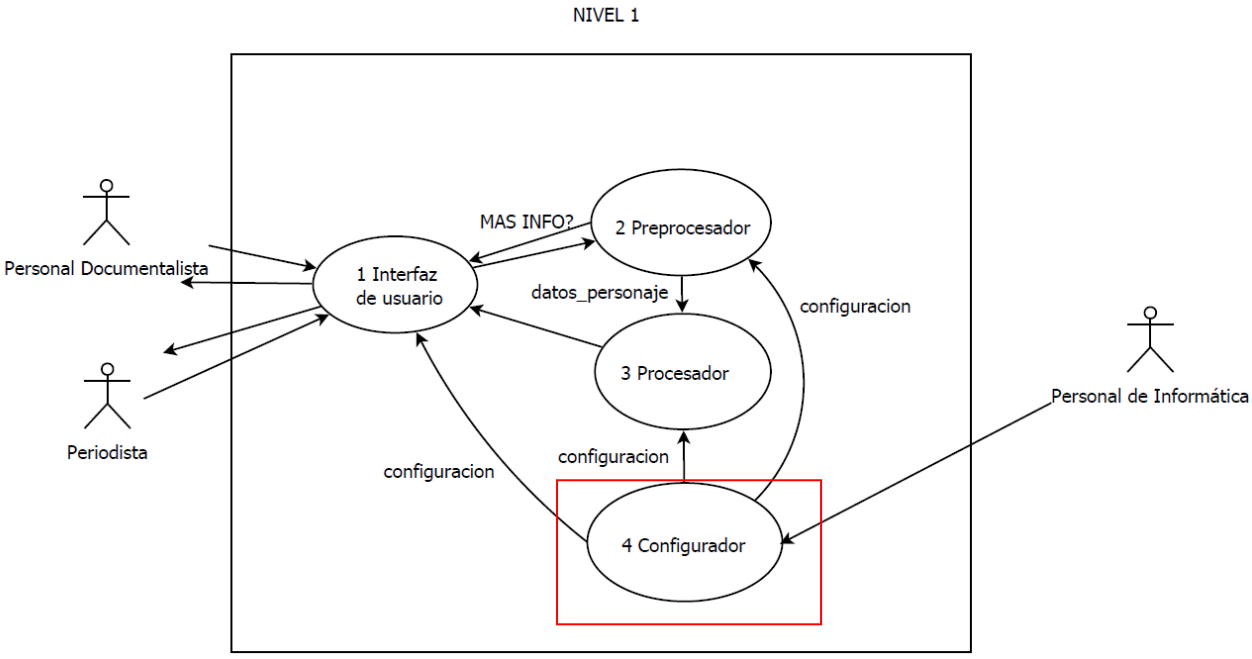

Figura 3.35 Diagrama nivel 1.

4.CONFIGURADOR

Descripción del proceso:	
Crea un archivo XML que utilizarán la interfaz de usuario, el preprocesador y el procesador para configurar los resultados y realizar las búsquedas.	
Entradas del sistema	**Formato**
Datos de identificación proporcionados por el usuario	Texto
Datos de configuración proporcionados por el usuario	Texto
Salidas del sistema	**Formato**
Configuración	Archivo XML
Recursos:	
BD Usuarios	Base de Datos
Subprocesos:	
Validación de usuarios	
Interfaz de entrada de datos	
Escritura del archivo XML de configuración final	

Estructura del fichero de salida configuracion en formato XML:

<XML>

<configuracion>

<Nombres_TablasYatributos>

<Prefijo_Tablas_fotos> Prefijo del nombre de las tablas de fotos (dbo.F_Fotosxxxx) **</ Prefijo_Tablas_fotos >**

<Prefijo_Tablas_textos> Prefijo del nombre de las tablas de fotos (dbo.T_Textosxxxx)**</ Prefijo_Tablas_textos >**

<Nombre_Tabla_catalogoNE> Nombre de la tabla que contiene el catálogo de Named Entities **</Nombre_Tabla_catalogoNE>**

<Nombre_Tabla_Tesauro> Nombre de la tabla que contiene el tesauro **</Nombre_Tabla_Tesauro>**

</ Nombres_TablasYatributos >

<**Pesos_busqueda_noticias_relevantes>**

Ruta y nombre del archivo que contiene los pesos que utilizaremos para elegir las noticias relevantes.

</**Pesos_busqueda_noticias_relevantes>**

<Cadenas_busqueda_Internet>

</Cadena_busqueda_facebook> Texto con el fragmento de URL que nos permite buscar el personaje en Facebook

</**Cadena_busqueda_facebook>**

<Cadena_busqueda_twitter> Texto con el fragmento de URL que nos permite buscar el personaje en Twitter

</**Cadena_busqueda_twitter>**

<Cadena_busqueda_linkedin> Texto con el fragmento de URL que nos permite buscar el personaje en Linkedin

</**Cadena_busqueda_linkedin>**

<Cadena_busqueda_google+> Texto con el fragmento de URL que nos permite buscar el personaje en Google+

</**Cadena_busqueda_google+>**

</ Cadenas_busqueda_Internet >

<Busqueda_DBpedia>

<Busqueda_DBpedia_español>
Texto con el fragmento de URL que nos permite buscar el personaje en DBpedia en español
</ Busqueda_DBpedia_español>
<Busqueda_DBpedia_ingles>
Texto con el fragmento de URL que nos permite buscar el personaje en DBpedia en español
</ Busqueda_DBpedia_ingles>
</ Busqueda_DBpedia >

<Configuracion_busquedas>
<Fecha_inicio_busqueda_fotos> Fecha a partir de la que se comienza a buscar fotografías en el archivo documental
</Fecha_inicio_busqueda_fotos>
<Fecha_fin_busqueda_fotos> Fecha hasta la que se busca fotografías en el archivo documental
</Fecha_fin_busqueda_fotos>
<Fecha_inicio_busqueda_noticias> Fecha a partir de la que se comienza a buscar noticias en el archivo documental
</Fecha_inicio_busqueda_noticias>
<Fecha_fin_busqueda_noticias> Fecha hasta la que se busca noticias en el archivo documental
</Fecha_fin_busqueda_noticias>

<Num_noticias_relevantes> Número de noticias relevantes extraídas del archivo documental que se quiere incluir en la ficha biográfica resultante
</ Num_noticias_relevantes >
<Num_noticias_recientes> Número de noticias recientes
extraídas del archivo documental que se quieren incluir en la ficha biográfica resultante **</ Num_noticias_recientes >**
<Num_fotos_relacionadas> Número de fotografías relacionadas con el personaje, extraídas del archivo documental que se quieren incluir en la ficha biográfica resultante
</ Num_noticias_relevantes >
<Num_fotos_primer_plano> Número de fotografías de primer plano del personaje, extraídas del archivo documental que se quieren incluir en la ficha biográfica resultante
</Num_fotos_primer_plano >

<Num_fotos_internet> Número de fotografías extraídas de Internet del personaje, que se quieren incluir en la ficha biográfica **</Num_fotos_internet >**
<Num_blogs> Número de fotografías de primer
plano del personaje, extraídas del archivo documental que se quieren incluir en la ficha biográfica resultante
</Num_blogs >
<Num_libros> Número de fotografías de primer
plano del personaje, extraídas del archivo documental que se quieren incluir en la ficha biográfica resultante
</Num_libros >
<Num_videos> Número de fotografías de primer
plano del personaje, extraídas del archivo documental que se quieren incluir en la ficha biográfica resultante
</Num_videos >
</Configuración_busquedas>

<Datos_ficheros>

<Ruta_archivos_xml > Ruta, incluyendo servidor y carpeta, donde se almacenarán los archivos xml que utiliza la aplicación**</Ruta_archivos_xml >**

<Prefijo_nombre_xml_resultado> Nombre del archivo XML resultante, con la ficha biográfica. A éste se le añadirá la fecha y el atributo Contador_instancias del objeto configuración, para evitar que se sobrescriban en el mismo archivo dos o más
peticiones que coincidan en el tiempo

</ Prefijo_nombre_xml_resultado >

<Prefijo_nombre_xml_datos_iniciales> Nombre del archivo XML con los datos del personaje introducidos por el usuario. A éste se le añadirá la fecha y el atributo Contador_instancias del objeto configuración, para evitar que se sobrescriban en el mismo archivo dos o más peticiones que coincidan en el tiempo

</ Prefijo_nombre_xml_datos_iniciales >

</Datos_ficheros>

<Gestor_BD>

Parámetros del objeto gestorBD utilizado para conectar con la base de datos EMMA. Utilizamos la clase DBODataAccess.vb del entorno .net

<Nombre_servidor>Ruta del servidor donde está la base de datos (EMMA)**</Nombre_servidor>**

<Nombre_BD> Nombre de la base de datos (EMMA)

</Nombre_BD>

<Nombre_Usr> Nombre de usuario para conectarse a la BD para realizar consultas

</Nombre_Usr>

<Passw_BD>Password para conectarse a la BD para realizar Consultas **</Passw_BD>**

</Gestor_BD>

</configuracion>

</XML>

B3.3 Preparación del contexto de trabajo.

Para el desarrollo del generador de fichas biográficas es necesario realizar unos trabajos previos a la aplicación en sí.

En primer lugar, crear el listado de entidades de nombre, a partir del cual se creará el catálogo. Se explica en esta misma sección de forma más detallada en el punto 3.3.1.

En segundo lugar es necesaria una selección de los lugares en la web donde buscaremos los enlaces a libros, blogs, vídeos, redes sociales, etc. En el prototipo se utilizó un servicio de Bing para las búsquedas que ya caducó. Para la herramienta final se han utilizado APIs[31] de Google con la que localizamos noticias, vídeos y libros.

Finalmente, en función de las pruebas que se realizaron con el prototipo, se debe realizar una selección de las consultas SQL para las búsquedas en el archivo documental, y de consultas SPARQL para las búsquedas en DBpedia.
Un ejemplo sencillo que utilizaremos en el código: consulta del nombre, fecha de nacimiento y un resumen de su biografía de un personaje concreto como Edgar Allan Poe:

```
 SELECT ?name ?fn ?abs  WHERE {
<http://dbpedia.org/resource/Edgar_Allan_Poe><http://dbpedia.org/ontology/abstract>
?abs .
<http://dbpedia.org/resource/Edgar_Allan_Poe><http://dbpedia.org/ontology/birthDate>
?fn .
<http://dbpedia.org/resource/Edgar_Allan_Poe><http://dbpedia.org/property/id> ?name
.
 FILTER ( lang(?abs) = "es" )
 }
```

B3.3.1 Creación de un catálogo de Named Entities. Algoritmo ampliado

Aquí entraremos en detalle en el algoritmo que se ha utilizado para crear el catálogo que se utiliza para realizar la desambiguación entre NE y la búsqueda de información.

Proceso:

1. Extraer de DBpedia la lista de todos los personajes. De todas sus propiedades elegir “name”.

2. Recorrer el XML obtenido de DBpedia. Para cada personaje:
   - Extraer el nombre completo (valor de la propiedad)
   - Almacenarlo

3. Extraer la lista de personajes del tesauro del medio periodístico.

4. Crear la tabla CATALOGO_NE en la que se enlazan las dos listas anteriores:
   - Para cada elemento del tesauro de textos cuyo campo KeyFD1 = PERSONAS,

[31] https://developers.google.com/api-client-library/dotnet/apis/

    - Buscar que KeyFD2 esté dentro de la lista extraída de DBpedia. Si está, crear un registro con ese personaje de esa lista y el campo id del tesauro.
    - Si no está, crear una entrada en el catálogo con el nombre e id del tesauro.
- Para cada elemento del tesauro de fotos cuyo campo thesad0 = PERSONAS,
    - Buscar que los campos thesad1 y thesad2 estén dentro de la lista de DBpedia. Si está en la lista:
        - buscar si ese personaje ya está en la tabla y añadir al registro el campo clave del tesauro.
        - Si no está en la tabla, crear el registro con ese personaje de la lista y el campo clave del tesauro.
    - Si no está en la lista de DBpedia, crear una entrada en el catálogo con el nombre e id del tesauro.

5. Completamos el catálogo con personajes que aparecen en los textos de los artículos y que no están ni en DBpedia ni en el tesauro.
   Para cada texto almacenado correspondiente a una noticia:
    - Ejecutar Freeling [10] sobre el texto:
        - Extraer Named Entities de él y almacenarlas
        - Almacenar lematizado verbo anterior y posterior (si existe)
    - Con el Gazetteer [11] que forma parte del entorno de EMMA, detectamos de un artículo las NE correspondientes a lugares geográficos y las eliminamos de la lista anterior.
    - Para cada una de las NE restantes:
        - analizamos los verbos anterior y posterior para ver si corresponden a acciones humanas (comparar con una lista elaborada previamente: nacer, morir, hacer), asignamos una puntuación si los tienen.
        - comprobamos si es un nombre compuesto de dos o más palabras. Asignamos una puntuación si los tienen.
        - Actualizamos el contador de cada NE por número de apariciones, y se suma a la puntuación anterior.
        - Eliminamos las NE con la puntuación por debajo de un umbral determinado.

6. Para cada uno de los personajes extraídos del punto anterior:
    - Buscarlo en el catálogo de Named Entities.
    - Si no está, almacenarlo como una nueva entrada.

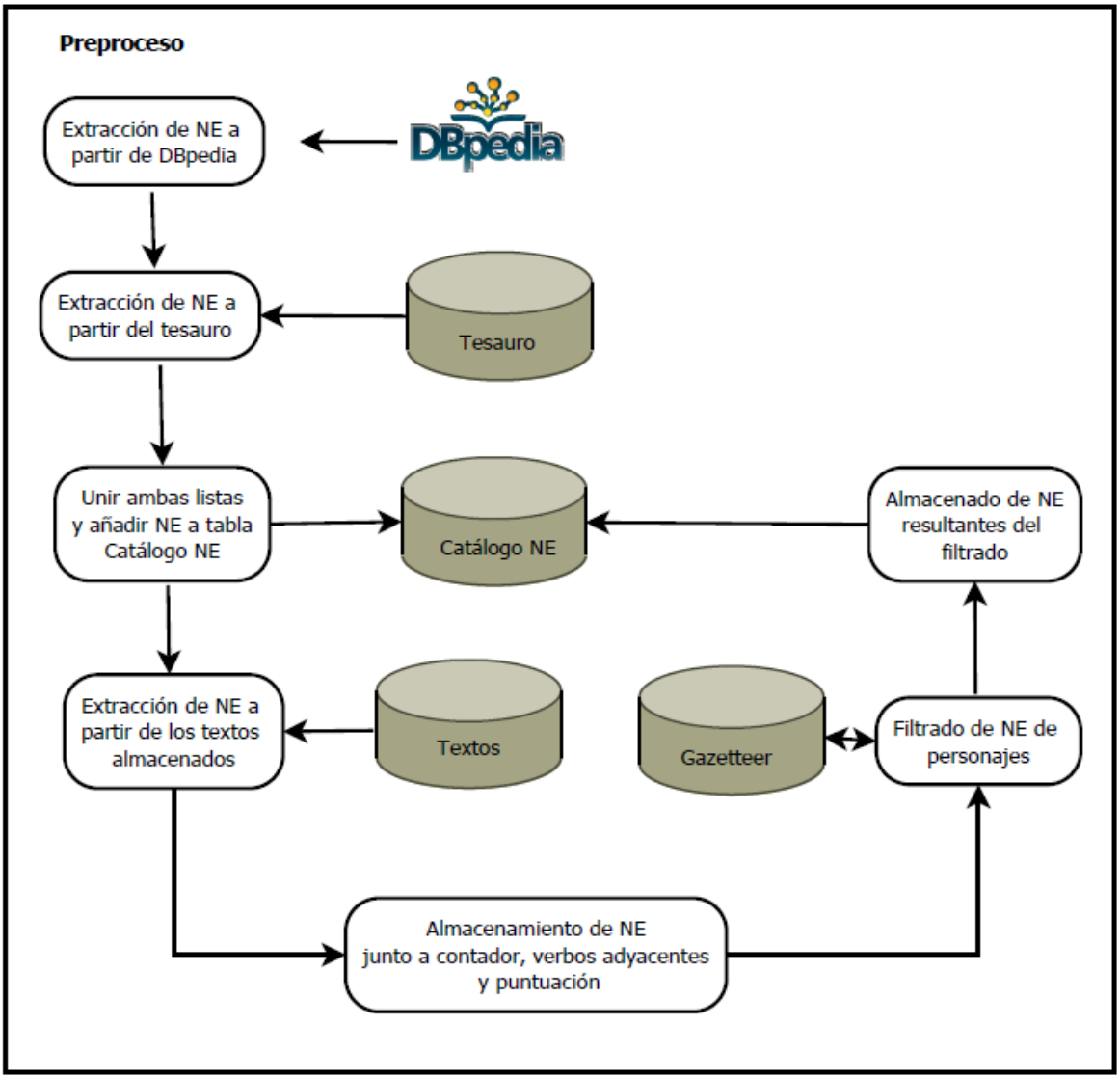

Figura 3.36: Diagrama de actividades ampliado del preproceso

El catálogo actual por limitaciones temporales, se ha realizado únicamente a partir de los personajes extraídos de DBpedia y los almacenados en el tesauro del Heraldo de Aragón. En total contiene 10.500 personajes.

El proceso ha consistido en seleccionar los recursos de DBpedia de tipo “Person”, lo cual devuelve un listado que hay que filtrar para extraer los nombres de cada personaje. Se ha podido ver que aun realizando diferentes consultas de SPARQL, el conjunto de entidades etiquetadas como persona en la DBpedia española es limitado, por lo que algunos personajes muy conocidos no aparecen (p.e. Mariano Rajoy). Los datos de la DBpedia deben transformarse para darle el formato que coincida con el del tesauro, para poder así enlazarlos. El resultado es una tabla que se ha tenido que limpiar “a mano” para obtener un catálogo que sea útil.

Después, con una periodicidad bimensual, este proceso se volverá a ejecutar de forma

incremental, añadiendo los resultados nuevos que van apareciendo en el tesauro. Este proceso será supervisado por el personal de documentación.

La herramienta para actualizar el catálogo actualmente es un ejecutable, que el personal de informática del Heraldo convertirá en un servicio Windows.

B3.3.2 Filtrado de fotografías de primer plano

En primer lugar, se ha visto la necesidad de etiquetar las fotografías, identificando los personajes que puedan salir en ellas.

Para ello, se hace que para cada personaje del catálogo de Named Entities se recorran todas las fotografías buscando en el tesauro y la descripción. Si están publicadas, se mira también en el campo del texto de la noticia correspondiente que almacena el pie de página.

Si aparece el personaje, se crea una referencia que enlace la fotografía con el catálogo de Named Entities y se actualiza el campo del tesauro de la fotografía.

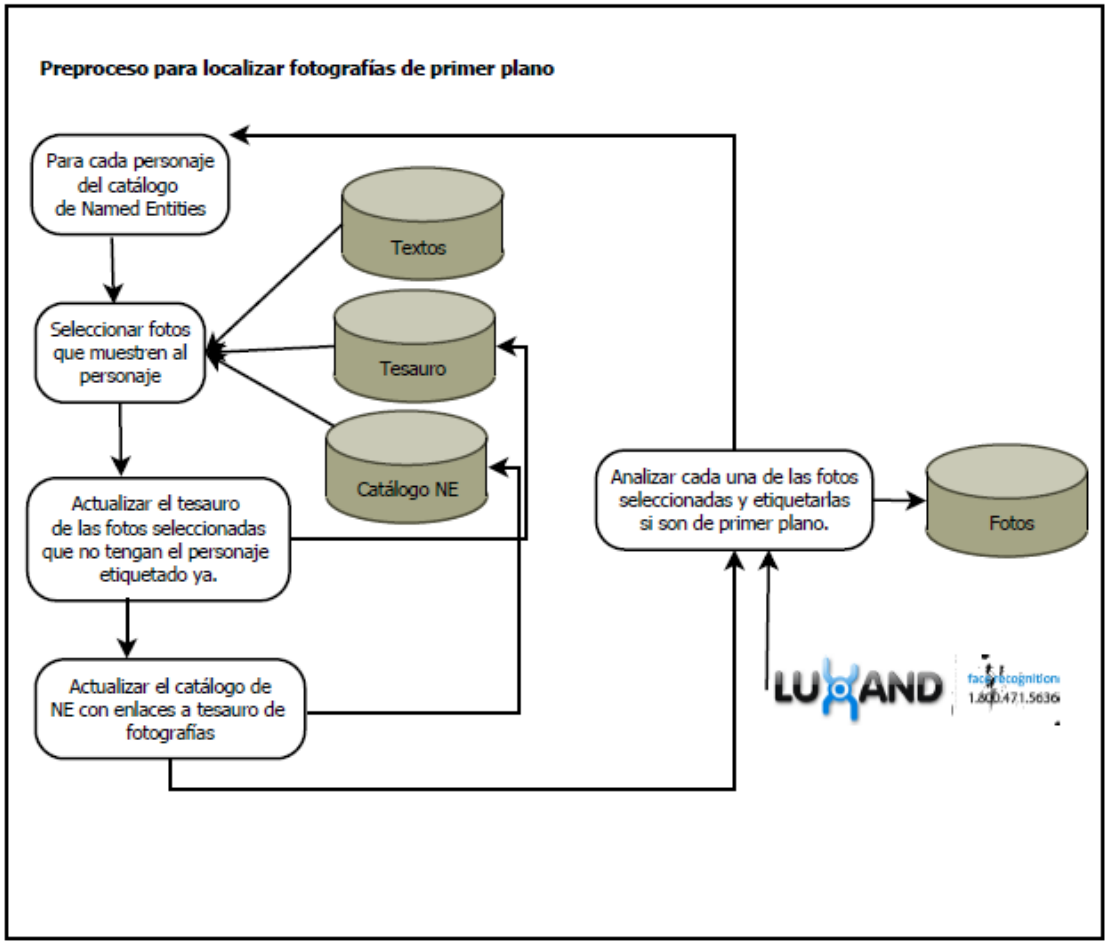

Fig. 3.37: Preproceso para fotografías de primer plano.

Para el segundo problema que es concluir si una foto es de primer plano o no, se han utilizado unas librerías desarrolladas por la empresa Luxand (ver en la memoria principal, capítulo 2.5), denominadas FaceSDK 5.0. Éstas permiten la detección de caras, devolviendo el tamaño de la imagen detectada. En realidad, la librería nos devuelve las coordenadas del centro de la cara y la altura y anchura de la cara. Con estas medidas se calcula una aproximación del área de la cara. El porcentaje de la cara sobre el área total de la fotografía (que también nos devuelve la misma librería) nos permite determinar si es un primer plano o no. Por tanto, se analiza con esta herramienta cada fotografía y la se etiqueta con el tipo de plano obtenido.

Para un primer plano, el porcentaje del área de la cara sobre el total de la foto se ha situado en un 20% o mayor. Si está entre el 10 y el 20% se considera un plano medio.

Estas librerías de reconocimiento facial se han escogido por haber sido utilizadas ya previamente en el Heraldo.

Después de probar la librería sobre un conjunto de 100 fotografías (elegidas al azar), el porcentaje de aciertos es muy bajo (20%). Sin embargo, si se escoge un conjunto que a priori se conoce que cada foto contiene al menos una persona (como sucede con el escogido en este proyecto, ya que se eligen fotos etiquetadas con personajes del tesauro) los aciertos suben del 90%.

B3.4 Métodos de desambiguación.

Tras los estudios previos (ver conclusiones al final de la sección Anexo A, punto 1.3.5), basándonos en el artículo de Brunescu [BUPA06], para los problemas de desambiguación entre dos nombres de entidades se han tomado las siguientes decisiones de cara al diseño:

1. Será el propio programa el que decida la entidad adecuada, dejando para un futuro la opción de que el usuario intervenga en la toma de decisiones.

2. Utilizaremos un catálogo previo de Named Entities, que contrastaremos con el tesauro y el campo “keywords” que acompaña a cada texto, para ayudarnos a identificar el nombre del personaje lo más correctamente posible.

3. Finalmente, utilizaremos la frecuencia de apariciones. Se realizará una búsqueda previa, donde se almacena el número de apariciones en el archivo documental. Si hay necesidad de desambiguar, buscaremos también apariciones en DBpedia español e inglés. El más frecuente, o en caso de empate, el de aparición más reciente, será el escogido.

B3.5 Sistema de pesos para seleccionar noticias.

Para poder elegir las noticias más relevantes, hemos tenido también que diseñar un algoritmo que nos dé los resultados más cercanos a los deseados. Para ello hemos utilizado información extraída de diferentes artículos de investigación, de los que destaca el propuesto por Luhn en 1958 [LUHN58], y del que sacamos la idea (algoritmo de ponderación basada en frecuencias de aparición de palabras y prefiltrado) para destacar una noticia sobre otras.

Se basa en que las palabras que más se repiten, son más relevantes excepto categorías léxicas cerradas (determinantes, pronombres, preposiciones, verbos auxiliares…). En este caso, utilizaremos la aparición del nombre del personaje.

En la figura 3.37 se resume el proceso para etiquetar la relevancia de una noticia respecto a un personaje concreto:

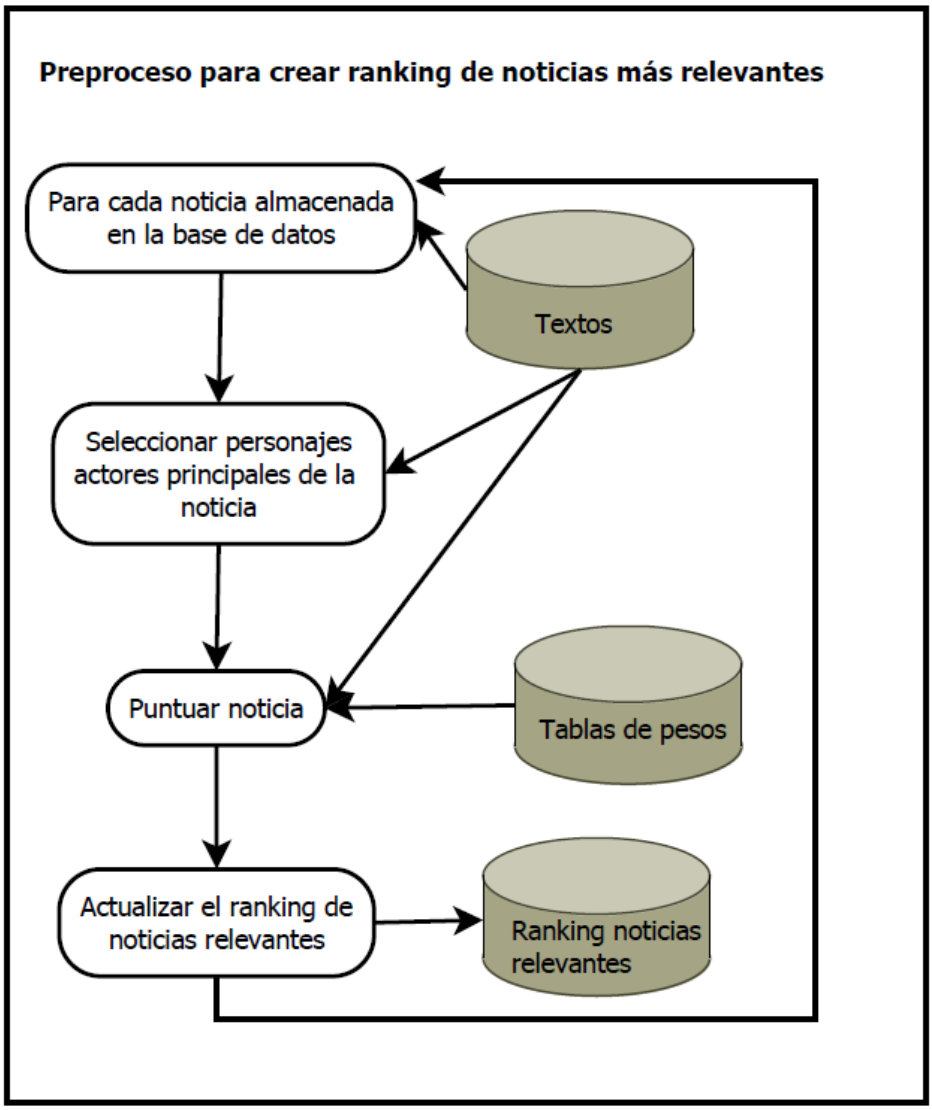

Fig. 3.37: Preproceso para valorar noticias más relevantes.

En primer lugar, contaremos el número de palabras de la noticia, seleccionando las que superen un determinado umbral. Puntuaremos las apariciones del nombre del personaje tanto en el título de la noticia, el pie de foto (si lo tiene), como en el texto, y en los campos *keywords* (palabras clave que describen el texto) y *resumen* de la tabla de textos.

El proceso crea unas tablas extraídas del fichero de configuración, que asignan determinados pesos para cada intervalo de apariciones del nombre del personaje en cada uno de los lugares “clave” de una noticia (se describen más adelante). Se puntúa

la noticia para cada uno de los personajes que aparecen identificados en el campo *keywords* del registro correspondiente a cada texto. Para cada uno de estos personajes se actualiza la tabla *Ranking* que se ha creado para en la base de datos EMMA, incluyéndola o no en función de que la puntuación obtenida sea mayor de las que ya están almacenadas. Para ajustar los resultados, se han realizado pruebas con diferentes pesos para determinar cuáles son los más adecuados para obtener las noticias deseadas.

Parámetros utilizados actualmente para valorar un texto:

1. Número de palabras de la noticia:
2. Número de apariciones del nombre del personaje en el texto
3. Número de apariciones del nombre del personaje en el resumen
4. Número de apariciones del nombre del personaje en el título
5. Número de apariciones del nombre del personaje en el pie de página
6. Número de apariciones del nombre del personaje en el campo *keywords*
7. El texto es portada
8. El texto aparece en página impar
9. El texto es contraportada
10. El texto es un monográfico

En la tabla que aparece a continuación, se indican los pesos actuales que se utilizan para valorar un texto (tabla 3.1):

PARÁMETROS	UMBRALES	PESOS
Nº de palabras del texto	0 a 50	0
	50 a 100	10
	>100	20
Nº apariciones en el texto	<1	0
	De 1 a 2	10
	De 2 a 9	20
	>9	30
Nº apariciones en resumen	<1	0
	1 a 2	20
	>2	30
Nº apariciones en título	<1	0
	>=1	30
Nº apariciones en pie	<1	0
	>=1	30
Nº apariciones en el campo *keywords*	= 0	0
	= 1	30
Es portada	Si	100
	No	0
Es página impar	Si	20
	No	10
Es contraportada	Si	70
	No	10
Es entrevista	Si	70
	No	10
Es monográfico	Si	100
	No	10

Tabla 3.1: Parámetros, umbrales y pesos para valorar una noticia.

Para un futuro, en vez de repasar todas las noticias en todos los casos, para personajes muy conocidos, se pueden utilizar umbrales ya que una portada rondará los 250-300 puntos, y una contraportada los 200- 260 puntos. Si suponemos que las portadas o contraportadas más recientes son las más relevantes, podemos reducir el tiempo de búsqueda.

El almacenamiento de estas noticias relevantes, se realiza en una tabla auxiliar de la base de datos, donde se registra para cada personaje del catálogo de Named Entities que tenemos guardado, un enlace a las N noticias más relevantes con su puntuación hasta ese momento.

En el momento en que se realizó el primer etiquetado, de los 2.000.000 de textos que están almacenados en EMMA, hay unos 110.000 que contienen descriptores para personas. El tiempo de puntuar y registrar los resultados (para un ranking de 10 noticias por personaje) es de 5 segundos para 100 noticias, y el procesado de todos los textos ha tenido un tiempo de 28 horas aproximadamente. Todo este proceso mejorará cuando se complete el etiquetado de noticias, que ahora ya se hace en todas las que se generan. Podría hacerse en un futuro para los textos antiguos no etiquetados buscando el personaje en otros campos como en el resumen del texto.

Si este ranking no se hubiera hecho, calculando que para Ramon y Cajal existen unos 52.000 artículos (incluyendo los no etiquetados), el tiempo de ejecución del generador de fichas biográficas aumentaría en unos 45 minutos, algo que para el usuario habitual es inaceptable.

Para cada edición del periódico, habrá que realizar el proceso de etiquetado de las nuevas noticias, para ver si se incluyen o no en el ranking. Esto se hará de manera independiente cada noche, para que no se vean afectados los tiempos de ejecución del generador de fichas biográficas. La herramienta para actualizar la puntuación de las noticias actualmente es un ejecutable, que el personal de informática del Heraldo convertirá en un servicio Windows.

B3.6 Diseño de la base de datos

A continuación se describen las tablas que forman la parte de la base de datos existente en el periódico y que se utilizan en la herramienta, y las añadidas para este proyecto.

En las dos primeras secciones se incluyen las tablas ya existentes en el medio periodístico y a las que se va a acceder para buscar información. En la tercera se describe la aportación de este PFC a la base de datos EMMA.

B3.6.1 Tablas para almacenamiento de textos

De este esquema, utilizaremos la tabla Textos y las tablas TColumna, TCuadernillo y TSección que utilizaremos en el sistema de pesos (ver Sección 3.4) para valorar la relevancia de un artículo según su posición dentro del periódico.

La estructura de tres últimas tablas incluye:
-TCuadernillo:
Empresa es clave ajena de empresa
Pub es clave ajena de publicación.
Edición es clave ajena de edición
-TSección:
Empresa es clave ajena de empresa
Pub es clave ajena de publicación.
-TColumna:
empresa, pub y edición son claves ajenas de empresa, publicación y edición.
-Thesaurus:
nTexto y clave thesaurus son claves ajenas de Textos y Thesadat.
-Thesadat:
KeyFDx contienen las palabras clave que forman el tesauro, y el campo Descri su descripción.

La tabla Textos es una de las más utilizadas en esta herramienta.

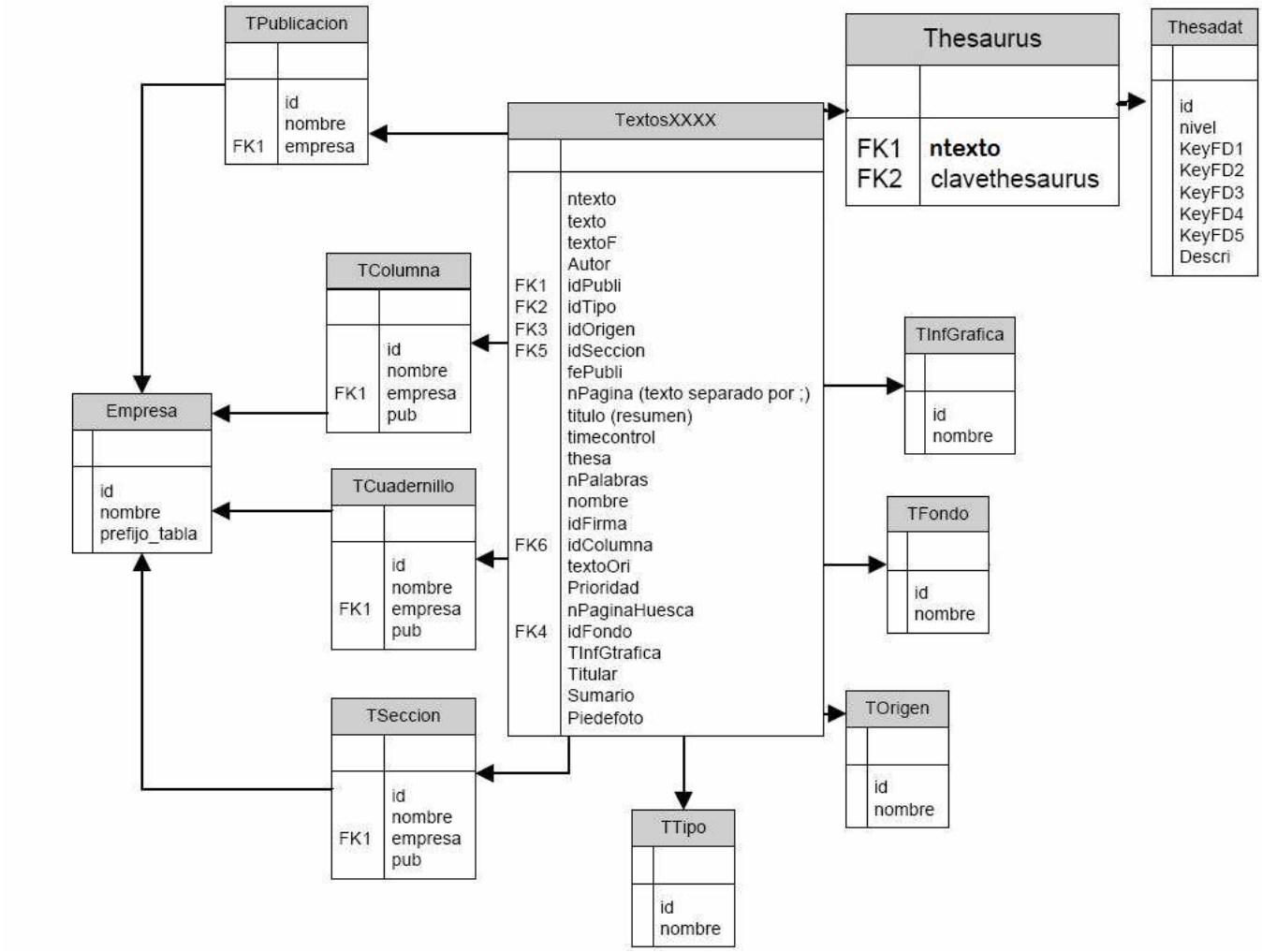

Figura 3.38: Diseño de las tablas relacionadas con los textos de los artículos.

A continuación se describen los campos de la tabla Textos:

Campo	Tipo dato	Descripción
Id	Entero	Clave del texto
Texto	Text	Texto que compone el artículo
TextoF	Text	XML que compone el artículo
Autor	Varchar(500)	Autor/es separados por ';'. Los apellidos separados por ','
idPubli	Short	Clave ajena a la publicación
idTipo	Short	Clave ajena al tipo
idFondo	Short	Clave ajena al fondo
idOrigen	Short	Clave ajena al origen
idSeccion	Short	Clave ajena a la sección
fePubli	Date	Fecha de publicación
nPagina	Varchar(90)	String con las páginas en cada edición. Ej:007,H007 significa que tiene la página 7 en la edición general y página 7 en la edición cuya letra es H. En nuestro caso, Huesca.
Titulo	Varchar(500)	Contiene el título del artículo
Timecontrol	-	No se usa.
Thesa	Varchar(500)	Contiene el tesauro del artículo
nPalabras	Entero	Número de palabras del texto
Nombre	Varchar(50)	Contiene el nombre del artículo
idCuadernillo	Short	Clave ajena al cuadernillo
idColumna	Short	Clave ajena a la columna
idInfGrafica	Short	Clave ajena a la información gráfica
Titular	Varchar(2000)	Contiene el titular si lo hay
Sumario	Varchar(2000)	Contiene el sumario si lo hay
Pie_de_foto	Varchar(2000)	Contiene el/los pie/s de foto si los hay

B3.6.2 Tablas para el almacenamiento de fotografías

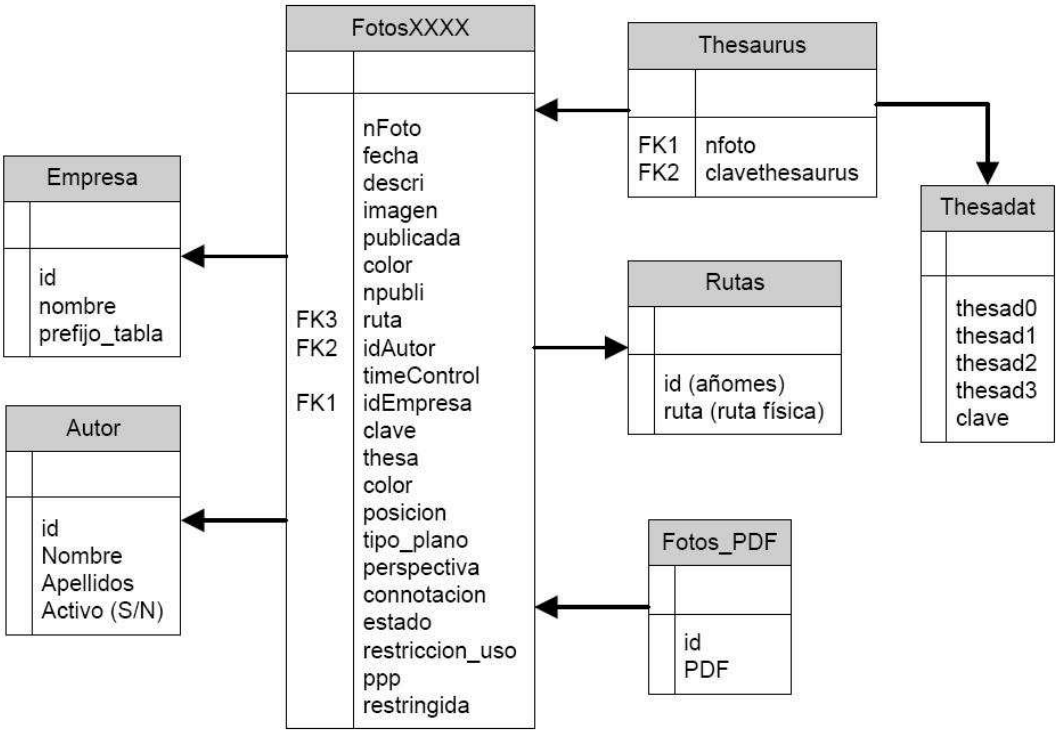

Figura 3.39: Diseño de las tablas relacionadas con las fotografías almacenadas.

Las tablas auxiliares (que contienen información adicional sobre las fotografías) son las siguientes:

-Thesaurus:
nFoto y clave thesaurus son claves ajenas de Fotos y Thesadat. Es la tabla que nos proporciona el o los personajes fotografiados.
-Empresa:
Nombre: contiene el nombre de la empresa propietaria de la fotografía.
-Autor:
Contiene los datos del autor y si sigue o no en activo.
-Rutas:
el campo ruta indica la ruta física en los servidores ZARSV002 y ZARSV003 donde se almacenan en carpetas los archivos binarios.
-Fotos_PDF:
PDF almacena el nombre del archivo PDF que contiene la fotografía.

Los campos de la tabla Fotos son:

Campo	Tipo dato	Descripción
restriccion_uso	short	clave ajena de restriccion_uso
restriccion_acceso	short	indica tipo de restricciones de acceso a la fotografía
publicada	bool	verdadero si la foto ha aparecido en alguna publicación
posicion	short	clave ajena de posicion
pixels	integer	número de pixels
perspectiva	varchar(50)	clave ajena de perspectiva
notas	varchar(2000)	anotaciones sobre la fotografía
nFoto	integer	clave principal
imagen	varchar(50)	nombre de la imagen
idAutor	integer	clave ajena de autor
id_propietario	integer	clave ajena de propietario
id_herramienta_grafica	short	clave ajena de herramienta grafica
fecha	date	fecha de publicación si la hay
estado	short	clave ajena de estado
descri	varchar(2000)	descripción de la fotografía
connotacion	varchar(50)	clave ajena de connotacion
color	bool	verdadero si la foto es en color
colección	varchar(50)	clave ajena de colección
clave	integer	identificador de la fotografía
anchura	integer	anchura de la fotografía
altura	integer	altura de la fotografía

B3.6.3 Tablas para el almacenamiento de Named Entities

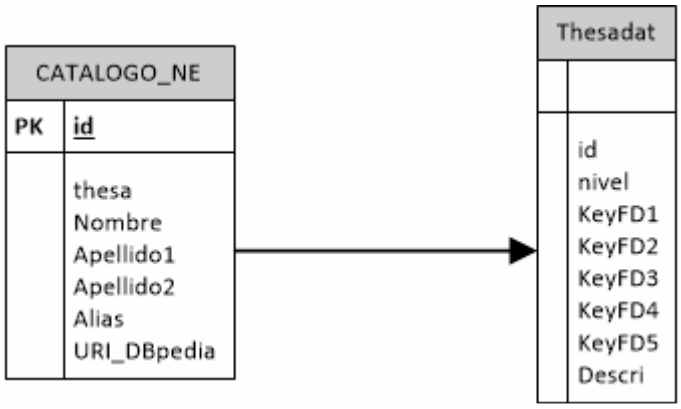

Figura 3.40: Diseño de las tablas relacionadas con el catálogo de Named Entities.

La tabla CATALOGO_NE contiene el campo thesa, que es clave ajena de la tabla Thesadat (que contiene el tesauro), y además el nombre de los personajes (Nombre, apellidos, alias y URI del personaje en DBpedia).

Campo	Tipo dato	Descripción
thesa	short	clave ajena de Thesadat
Nombre	varchar(50)	Nombre del personaje
Apellido1	varchar(50)	Primer apellido del personaje
Apellido2	varchar(50)	Segundo apellido del personaje
Alias	varchar(50)	Alias del personaje
URI_DBpedia	varchar(100)	URI de la entidad (personaje) en DBpedia

B3.6.4 Tablas para el almacenamiento de la valoración de las noticias

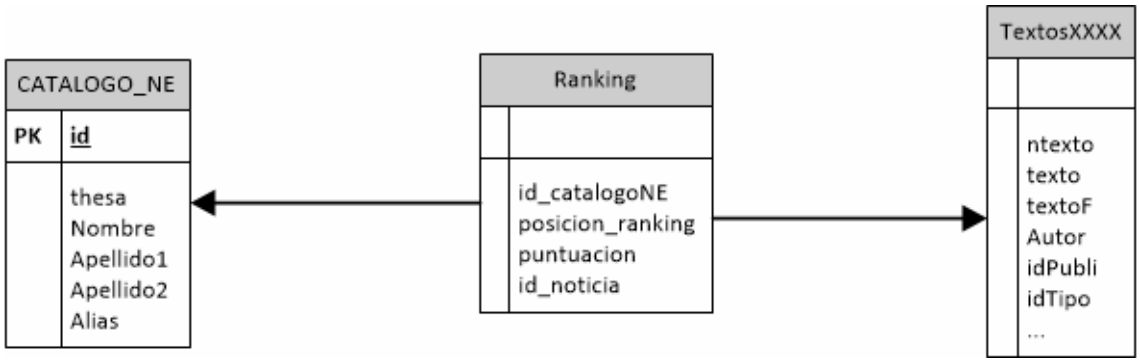

Figura 3.41: Diseño de las tablas relacionadas con la puntuación de noticias.

La tabla Ranking contiene el campo id_catalogoNE que es clave ajena del personaje dentro de la tabla CATALOGO_NE, el campo id_noticia que es clave ajena de Textos identificando una noticia relacionada con el personaje de su mismo registro.El resto de campos: posición_ranking indica para un personaje concreto, qué posición ocupa en el ranking la noticia almacenada en ese registro. El campo puntuación es un campo numérico que almacena el valor alcanzado por la noticia en el proceso de puntuar su relevancia. Este proceso se describe en el algoritmo del capítulo 3.4, Anexo B.

Campo	Tipo dato	Descripción
Id_catalogoNE	short	clave ajena de CATALOGO_NE.
Posición_ranking	integer	Puesto en el ranking de la noticia para un personaje.
puntuación	integer	Puntuación obtenida al valorar la noticia.
Id_noticia	short	clave ajena de TextosXXXX

B3.6.5 Tablas de gestión

Esta sección hace referencia a todas aquellas tablas de datos que se utilizan para gestionar otros aspectos de la aplicación. Estas tablas ya existen y están documentadas y funcionando en las aplicaciones desarrolladas para la empresa. Éstas cubrirán distintas necesidades de la aplicación.
Entre estas tablas encontramos:
- Tablas de usuarios.
- Tablas de permisos.

B3.7 Módulo de consulta a DBpedia

En un primer momento, se realizó el acceso a la información de DBpedia a través de una librería escrita en Java, a la que accedía la librería principal. Esta librería llamaba a un servicio web diseñado en JAVA que lanzaba las consultas SPARQL contra DBpedia. Sin embargo, más adelante apareció un módulo en .NET desarrollado posteriormente y por coherencia con el resto de la codificación, se incluyó sustituyendo al primero. Así que el acceso a DBpedia se realiza en la versión definitiva a través de un ejecutable creado en .NET que interpreta la consulta SPARQL y nos devuelve un archivo XML con los resultados.

Un ejemplo de consulta básica a DBpedia, para recuperar la información sobre un personaje concreto es:

```
SELECT  ?y ?z
 WHERE {<http://es.dbpedia.org/resource/Federico_García_Lorca>  ?y ?z      }
```

En la que recuperamos todos los campos y sus valores del recurso (personaje) Federico García Lorca.

Respecto a la creación del catálogo de Named Entities, también hemos utilizado este módulo para recuperar una lista de todos los personajes almacenados en DBpedia (debemos recordar, que este repositorio está basado en la información guardada en Wikipedia). La consulta que utilizamos es:

```
SELECT  ?x WHERE {?x ?y <http://dbpedia.org/ontology/Person>  . }
```

Que nos devuelve todos los elementos que están catalogados dentro de la ontología como personas.

Estos resultados que obtenemos en los archivos XML se han filtrado para obtener o bien los nombres en el formato que almacenaremos en el catálogo, o bien otros datos que hemos incluido en la biografía.

B4. DESARROLLO Y PRUEBAS

En el desarrollo de la aplicación ya se ha mencionado en la memoria principal la metodología aplicada y la realización del primer prototipo (ver sección 2.3 de la memoria principal).

B4.1 Trabajo previo

El primer paso fue la creación del catálogo de Named Entities, que se explica en la sección 2.4 de la memoria principal y en la 3.3.1 del Anexo B. Este catálogo permitió que se llevaran a cabo otra tarea previa al desarrollo de la herramienta en sí, que es la puntuación de cada una de las noticias y su inclusión en el ranking de noticias relevantes para cada personaje.

La valoración de la relevancia de una noticia es algo subjetivo, y conllevaría la lectura de todas las noticias relativas a un personaje concreto. Por ejemplo, tenemos 52.000 textos almacenados en el Heraldo de Aragón que hablan sobre Ramón y Cajal. Por eso el algoritmo de evaluación se ha hecho en consenso con el departamento de documentación a la hora de escoger los ítems que nos pueden aportar datos sobre el valor de un artículo (ser portada, estar en página impar, número de palabras, etc.). También la evaluación de los resultados se ha hecho conjuntamente, ajustando los pesos para cada ítem hasta que el documentalista encargado de probar la herramienta ha aprobado los resultados.

El siguiente paso fue el etiquetado de las fotografías, que fue llevado a cabo por el personal del Heraldo utilizando la herramienta desarrollada en este PFC. En este caso, se han obtenido entre un 60 y 70% de aciertos para fotografías de cualquier tipo, y más de un 90% si se escogen sólo fotografías de personas.

Así quedaron completos los campos *puntuación_noticia* para cada texto, *tipo_plano* para cada fotografía, y se llenaron las tablas *CATALOGO_NE y Ranking.*

B4.2 El generador de fichas biográficas

La implementación del generador de fichas biográficas se ha realizado en diferentes etapas, ya que la estructura favorece esto, según los tipos de búsqueda: en primer lugar se realizó toda la búsqueda en el archivo documental. Importante ha sido el diseño de las consultas SQL para conseguir un tiempo de acceso a los datos que no sea excesivo para el usuario.

Otra curiosidad acerca del acceso al archivo documental es la forma de organizar los textos y las fotografías. Por ejemplo, la tabla fotosXXXX es un conjunto de tablas denominadas así por el año que aparece en la fecha de la fotografía. Así, comienzan en fotos1800 hasta fotos2014, aunque no se almacenan año por año hasta 1990. Las primeras tablas almacenan fotografías del primer siglo (1800-1900) y las siguientes

cada 20 ó cada 10 años.

Esto se ha tenido en cuenta a la hora de recorrer todas las tablas, ya que los artículos sí son un conjunto con una numeración correlativa, pero en este caso no. Y además se ha hecho una búsqueda que sea fácil de adaptar al almacenamiento de cada publicación, ya que no es lo mismo en el Heraldo de Aragón que en el de Huesca, por ejemplo, en el que las fotos no se remontan tan atrás.

El segundo módulo desarrollado fue el de desambiguación, que utiliza la búsqueda en el archivo documental, y para el que se tuvo que desarrollar también la librería de búsqueda en DBpedia (ver sección 5.4).

A continuación, se creó el módulo que presenta los resultados, transformando el archivo XML que contiene los resultados en un HTML que pueda mostrar el navegador del usuario.

Finalmente, se amplió la ficha biográfica con otras búsquedas en Internet: libros, blogs, vídeos, páginas web personales y redes sociales en las que aparezca el personaje. También se añadieron datos extraídos de DBpedia en español.

Durante todo el desarrollo del proyecto se han ido realizando pruebas para comprobar la correcta funcionalidad del sistema. Estas pruebas han sido de cada parte por separado y también de todas ellas en conjunto.

El último paso fue la integración en el entorno de trabajo. De momento, se ha instalado de manera local en los puestos de los usuarios, pero se prevé una integración futura en la plataforma web de EMMA.

B4.3 Pruebas unitarias

Se ha verificado que todos los componentes del sistema, de manera individual, funcionan tal y como se espera. Para ello se han realizado acciones controladas, en las que se ponían a prueba tanto a los componentes debiendo identificar una situación de error y manejarla como devolviendo el control al usuario.

Pruebas en los formularios de entrada:

Respecto a los formularios de entrada detallamos pruebas para cada uno de los tres que se utilizan:

1. Búsqueda sencilla

El formulario debe cumplir los siguientes requisitos:

- Comprobar que el sistema informa correctamente sobre los campos obligatorios.
- Comprobar que no se distingue entre mayúsculas y minúsculas.
- Comprobar que la introducción de comillas u otros símbolos genera una solicitud de nueva entrada por parte del usuario.

Las comprobaciones se han realizado correctamente.

2. Búsqueda avanzada

El formulario debe cumplir los siguientes requisitos:

- Comprobar que el sistema informa correctamente sobre los campos obligatorios.
- Comprobar que no se distingue entre mayúsculas y minúsculas.
- Comprobar que la introducción de comillas u otros símbolos genera una solicitud de nueva entrada por parte del usuario.
- Comprobar que el sistema detecta e informa correctamente sobre los campos tipo fecha.
- Comprobar que el sistema detecta e informa correctamente sobre los campos numéricos.

Las comprobaciones se han realizado correctamente.

3.Configuración

El formulario debe cumplir los siguientes requisitos:

- Comprobar que el sistema informa correctamente sobre los campos obligatorios.
- Comprobar que no se distingue entre mayúsculas y minúsculas.
- Comprobar que el sistema detecta e informa correctamente sobre los campos tipo fecha.
- Comprobar que el sistema detecta e informa correctamente sobre los campos numéricos.

Las comprobaciones se han realizado correctamente.

Operaciones sobre componentes de los formularios:

Los componentes que forma parte del formulario han sido comprobados en su funcionamiento:

- Se ha comprobado que no se producen errores al pulsar sobre el componente.

- Se ha comprobado que al pulsar la tecla *Tab* el cursor se desplaza correctamente en el orden deseado por los campos.

Pruebas durante la generación de fichas biográficas

Se ha comprobado que es correcto el funcionamiento de los siguientes comportamientos del sistema:

- Si el personaje no aparece en la búsqueda previa, no realiza la ficha y el sistema avisa de lo sucedido.

- Que el proceso de desambiguación ofrezca siempre un único resultado que corresponda a un personaje recogido en el catálogo de Named Entities.

-Si no se puede conseguir la información correspondiente a alguna sección concreta de la ficha biográfica, ésta aparezca en blanco y no produzca ningún error.

- Comprobar que se eliminan símbolos extraños y problemas con la acentuación en los archivos XML y HTML resultantes.

B4.4. Pruebas de integración

Las pruebas de integración se han realizado encajando todos los subsistemas de la arquitectura empleada. Estas pruebas se han ido realizando a lo largo de la construcción de la aplicación, conforme se iban añadiendo funcionalidades a la misma. Se trata de comprobar que las diferentes capas de la arquitectura de la aplicación interaccionan entre sí correctamente. Para ello, se han realizado pruebas estructurales y funcionales.
Las pruebas estructurales se centran en las llamadas entre los diferentes procedimientos de las diferentes capas de la arquitectura. Se realizan identificando todos aquellos posibles casos de uso para poder comprobar que su funcionamiento es el correcto.
Las pruebas funcionales se centran más en encontrar fallos en los módulos cuyas respuestas dependen de las llamadas a otros módulos.

B4.5. Pruebas del sistema y aceptación

Una vez probado todo por separado, se ha probado el sistema para comprobar el correcto funcionamiento del mismo bajo distintas circunstancias: entradas erróneas de datos, fallos, etc.

Finalmente se han revisado uno por uno todos los requisitos definidos en la fase de análisis para comprobar que han sido satisfechos.

ANEXO C: ARTÍCULO DE INVESTIGACIÓN

C.1 “Generating automatic data sheets from mixed environments - Experience in a Media Company (Experience paper)”

A continuación se incluye el borrador del artículo de investigación desarrollado con el grupo SID, en cuya elaboración he participado junto a Angel L. Garrido, María G. Buey y Sergio Ilarri. El artículo se encuentra en la categoría de *Experience paper.*

El día 1 de diciembre se presentará la redacción definitiva a la 27ª Conferencia Internacional CAISE[32] 2015 (Internacional Conference on Advanced Information Systems Engeneering, CORE A) que se celebrará en Estocolmo en Junio de 2015.

[32] http://caise2015.blogs.dsv.su.se/

Generating automatic data sheets from mixed environments - Experience in a Media Company

Angel L. Garrido, Pilar Blazquez, María G. Buey, and Sergio Ilarri

IIS Department, University of Zaragoza, Spain
{mgbuey,garrido,silarri}@unizar.es

Abstract. Nowadays, there is a huge amount of digital data stored in private repositories with valuable information that are unreachable from Internet. Only small groups of people are allowed to access to these databases in order to work or to research. This closed and isolated repositories, however, can be enhanced in a remarkable manner when combined with the information provided by the Web from multiple repositories free to use. In this way, the user could have a better experience, save time and improve the quality of the recovered information.
In this paper we present a creative experience work in the field of Engineering Systems. The main objective of this project is to develop an application that can automatically generate structured information sheets. We use the texts of the news, the newspaper pages, and the photographs belonging to a document database of a media, but we also use other Internet sources, including social media pages and linked data. We have integrated it into an existing CMS in a real Media Group organization and we have tested it to evaluate its effectiveness. Results achieved are hopeful and show the interest of the approach.

Keywords: information extraction; databases; Internet; linked data;

1 Introduction

When searching over large information repositories, it is usual to get a lot of results that, subsequently, must be filtered manually in order to obtain the desired information. This happens for example in the web, but the problem is beginning to happen in companies and public organizations that store private and large amounts of information. This is a major problem, since the time spent looking information represents a significant loss of time and resources. Moreover, the results of performing a manual filter on a significant amount of data are be very limited. This leads us to seek solutions to locate and condense information more efficiently. This issue today is still an open problem since often the information is unstructured, usually it is in the form of natural language, which hinders even more its processing and its identification.

But when workers, researchers or users in general access to these private databases of documents, they have an additional problem: these documents usually are isolated. Perhaps the data is related with other internal repositories,

but they are not connected with the Web. And this lead the users to make and additional work if they want to mix these private information with the public information that everybody can obtain in the Web.

This situation can be found for example in big companies, research centres, public administrations, big libraries and media companies. Our case of study is Grupo Heraldo[1], a major media company in Spain including written publications and audiovisual business. This company owns several local newspapers with more than 120 years of regional information. Therefore, the company owns a valuable amount of specific information about some Spanish regions: news, photographs, interviews, reports, and multimedia data like animations, web pages, graphics or videos. All these information is stored in a relational database with over 10 millions of registers. The whole repository is closed to the general public, and it is only accessible to documentalists, journalists and researchers. This system, called EMMA, is a content management system specific for media companies developed by Hiberus Software[2].

The journalistic job is to identify and to investigate issues of public interest. They search for information, they contrast this information with other sources, and finally they synthesize all this information in a document in order to publish it. For this task, journalists may need information previously published. In any media company, this information is stored in the archive, and this archive is managed by documentalists. The work of the documentalist nowadays is complex due to the large amounts of information handled. Everyday, they must store and classify lot of news. And when a journalist required them an information, documentalist have to search for it all over the archive, typically populated with millions of registers. The EMMA CMS system provides to users a good set of tools in order to search, to filter, and to reach the desired information, and the system includes some advanced techniques related to Natural Language Processing (NLP), machine learning and ontologies in order to help the documentalists in their labours of categorising and tagging, as can be seen in [1–3], but at the end it is a closed system. Therefore, when users search information about a town, an event, a company or a person, they must complete the sheet by searching in other sources, for example, in the Web.

In this work we show an experience work in a real company of the Heraldo Group: Heraldo de Aragón[3]. The goal of this project is to develop a software integrated with EMMA CMS that will be able of automatically generate informative sheets about famous people. For doing this, we will adopt the role of a typical user that must to search the most suitable results in the private database (news, newspaper pages, photographs, etc.), and later this information must be enriched with data obtained from the Web. The adventage of this project is pretty clear: Saving time on the daily work of journalists and documentalists. Search among items and archived photographs carries a lot of time and effort, and the staff of company have other more valuable tasks to perform.

[1] http://www.grupoheraldo.es
[2] http://www.hiberus.com
[3] http://www.heraldo.es

For doing this enrichment work from Internet, we will use any internet sources, including social media pages and linked data. We found that a good sample of useful linked data to apply in this project could be DBpedia[4]. DBpedia [4] is a project for the extraction of data from Wikipedia and for transforming them into a semantic repository. The DBpedia project is conducted by the University of Leipzig, Free University of Berlin and OpenLink Software company.

Before starting, we have found some initial difficulties that require special attention:

- Extract specific information from a large number of texts in natural language with a journalistic writing style is not an easy task. There are a many studies that have faced this problem, and it is not solved yet.
- Another difficulty arises in recovering suitable pictures of the characters. Currently there are near 3,000,000 photos in EMMA. Although the database scheme provides a field which defines the type of plane, these labels are mostly blank. Therefore, if we want to add a close-up photo of the character we must determine whether the photographs is a really close-up by using algorithms implemented using a external tool called FaceSDK, developed by Luxand[5].
- The system must unambiguously solve conflicts between two characters with the same name or between a character and a name of another kind of entity. For example, if we seek information about Michel York and the user does not give the name, all the information related to York as a city must be manually filtered. For helping the resolution of ambiguity between characters, we have decide to create a catalogue of Named Entities [5]. This catalogue links the documentalists' thesaurus tags with DBpedia entries and, at the same time, with specific keywords found in the news that could represent particular persons.
- Finally, the last highlighted difficulty is to extract information from a Linked-Data based repository like DBpedia. While extracting information from structured sources may seem simple, we found some problems using DBpedia, for instance: bad integrity of the data, or the lack of concrete fields in some registers. These troubles have led us to look for characters in DBpedia using various languages in order to retrieve information as best as possible.

Some tasks have to be performed before the execution of the generator biographical sheets: It was necessary to add to the database the information about frequencies of keywords. These keywords have been previously obtained by using both the well-known algorithm TF-IDF[6] and Freeling[7], a specific software to process texts in natural language.

The execution time constraint also leads us to decide to prepare a pre-process that tags all the news, assigning a score to each person. Thus, the generator of biographical sheets directly accesses most relevant news for each character in a faster way.

[4] http://dbpedia.org/
[5] http://www.luxand.com/facesdk/

The main contributions of this experience work is the development of an Engineering System able of generating character sheets, which facilitates recovering the most important information about a person (for example, a politician, a footballer or a singer) from a database of semi-structured documents, but with the ability of adding information obtained from the Web: blogs, web pages, social media, linked data, etc. The idea of developing this automatic generator biographical records arises from the need for documentation when transmitting data to reporters about a particular character. We have used existing techniques related with information extraction, text mining, natural language processing (NLP) and semantics. We have contributed to state of the art by analysing an experience of application of all these techniques, and we have developed a new approach to tackle the typical problems related to these kind of systems, for example the disambiguation between items with similar names.

The rest of this paper is structured as follows. Section 2 explains the general architecture of the proposed system. Section 3 refers to the final implementation of the software. Section 4 discusses the results of our experiments with real data. Section 5 analyses other related works. Finally, Section 6 provides our conclusions and future work.

2 Methodology

In this chapter we make an overview of the tool, to provide a description that allows a comprehensive understanding of its performance. The main objective of this project is to develop a system that can generate automatically the information sheet of a character from the news that belong to the document database of a media and data that can be extracted from Internet. The system is responsible for generating electronic documents that contain related information with the selected character. This information consists of a selection of texts and relevant and present photographs, biographical information, links to websites, videos, social networks or links to other relevant information appearing on the Internet.

Search times have been very important when making design decisions, since the format of the SQL queries and the need of choosing the most relevant news, could lead to the need to create a tool for doing search in real time or creating previously a repository of information about the most important characters.

It has been necessary to develop previous works before the implementation of the generation of the character sheets. These works perform the following tasks:

- Generating a Named Entity catalogue of characters linked with the thesaurus tags of the company, in order to locate them on news and pictures. This task is explained in more detail in subsection 2.1.
- The close-up photos to be displayed in the biographical sheet are not properly labeled in the database. Analyzing each photo to search the character would take too much time, so we have developed a photo-tagging tool that is also described in subsection 2.2.

– Restricting execution times also leads to decide prior labeling of the news, assigning a score to each. This process is explained in subsection 2.3. Thus, the generator of biographical records directly accesses the "ranking" most relevant news for each character.

2.1 Creating a catalogue of Named Entities

This catalogue is used in the biography sheet generator to help in the searches for text and photos, and in the disambiguation of characters. The process (Fig. 1) to generate it consists of several steps:

1. Extracting a list of characters from DBpedia [4]. For each character it extracts its complete name and it stores its information in the database.
2. Extracting the list of characters from the thesaurus of the media company.
3. Creating the Named Entity catalogue where the process links both lists of characters. In order to carry out this step, for each element identified as a character in the the thesaurus, the process searches if it appears in the list of characters extracted from DBpedia. If it appears, it creates a new register with the information of that character and links it with its correspondent thesaurus, else the process only creates a new register with the thesaurus.
4. Completing the Named Entity catalogue with characters that do not appear in both lists, but they appear in the texts of the news. So, for each piece of news, it uses Freeling [7] to identify Named Entities and it stores them. It also stores the previous verb and the next verb that appears next to a Named Entity, if they exist. After that, the process uses the Gazetteer to detect Named Entities that correspond to geographical places and it eliminates then from the list. Next it repeats the previous task by using a list of companies, organizations and political parties.

The algorithm is completed using a Gazetteer and different lists of other entities because the Freeling tool has a labeling module that theoretically detects NE of people, but in practice this labeling is not reliable. This gazetteer is built from the GeoNames [6] geographical database. This data processing is performed completely on the entire set of information the first time that the catalogue is created. After that this process will run incrementally, adding new results that may appear.

We have incorporated the catalogue as a table in the database. Besides, the catalogue has let us to standardize the naming of characters that allows to locate characters more accurately inside (intranet) and outside (internet) of the documentary environment in which it works.

2.2 Tagging close-up photos

The techniques of facial development has become a regular part of commonly used tools, such as Facebook in the labeling of photos.

[6] http://www.geonames.org/

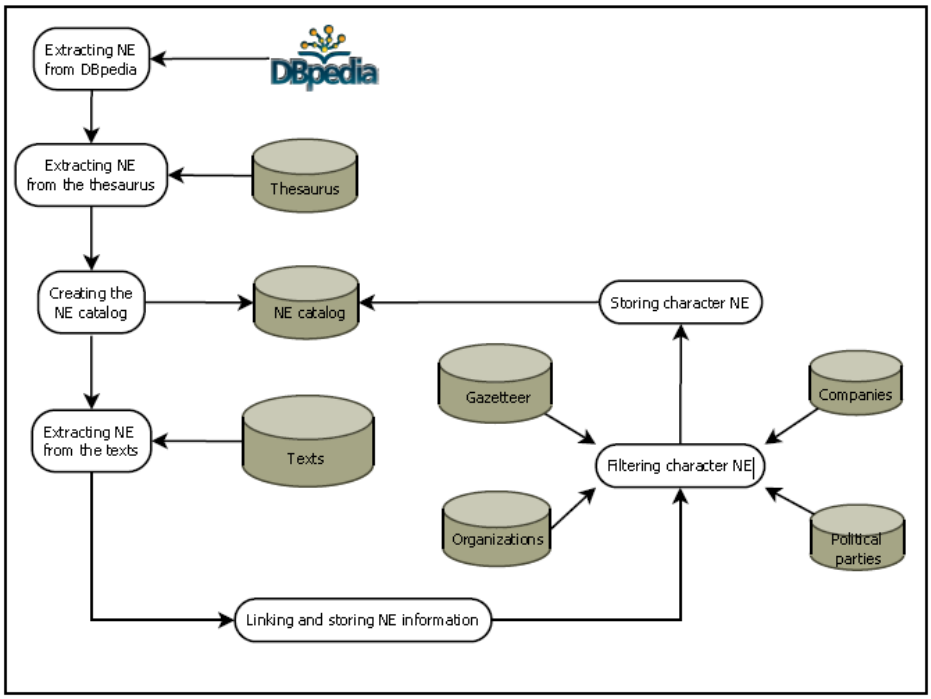

Fig. 1. Named Entity catalogue generation process.

If the biography sheet generator tries to determine the character and type of plane of each photograph obtained from a search in real time, the execution time rises excessively. For example, the character 'Jose Luis Rodriguez Zapatero' appears frequently in newspapers, only in the year 2010 there is 218 pictures whose description contain that name, and only 20 of them are labelled with the name of that person in the thesaurus. For one year we have nearly 240 photographs to perform the facial recognition to determine whether they are close-up or not, and the total time to locate photographs only takes 15 minutes.

Therefore, we face the need of labeling photographs previously, both for the characters and the type of plane they contain.

This process is shown in Fig. 2. For labeling the characters, each character which have been include in the catalogue of Named Entities described before, this process analyzes all the pictures looking for in the thesaurus and in the description of the pictures. If they are published, it also looks into the text field of the corresponding news that stores the footer.

If the character appears, a reference is created to link the picture with the catalog of Named Entities and the field of photography thesaurus is updated.

For the second problem, which is to conclude if a picture is close-up or not, it has been used the libraries of Luxand called FaceSDK 5.0, used in other

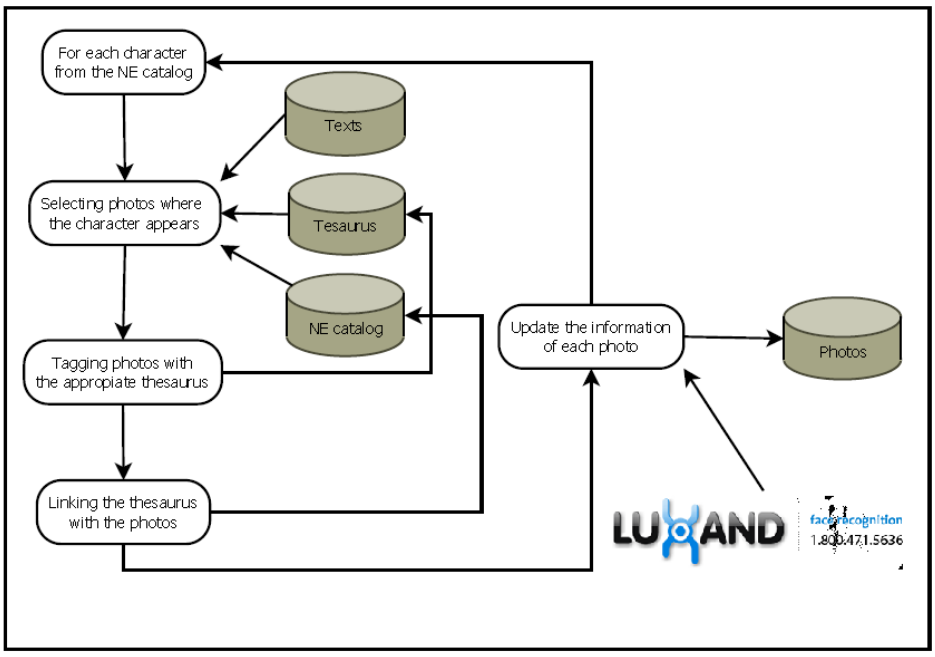

Fig. 2. Tagging close-up photo process.

works like [8, 9]. They allow the detection of faces and they return the size of the detected image. The percentage of the face taking account the total area of the photo allows us to determine whether or not the picture is a close-up. Therefore, it analyzes each image with this tool and labels it with the type of plane obtained.

2.3 Developing a ranking of relevant news

In order to choose the most relevant news, we have also design an algorithm that gives the closest desired results. For this purpose we have used information from different approaches, the most remarkable is [10], whose weighting algorithm based on frequency of occurrence of words and pre-filtering we have used to highlight pieces of news from others. It is based on the most repeated words are more relevant, except closed lexical categories (determiners, pronouns, prepositions, auxiliary verbs ...). In this case, the appearance of the character name is used.

First (see Fig. 3), the number of words in the news is taken account by selecting those that exceed a certain threshold. Occurrences of a character name are scored taking account the title of the story, the caption, the text and the abstract and keywords fields of the piece of news. It also consider whether the character appears on the front cover, back cover, odd page or a monograph.

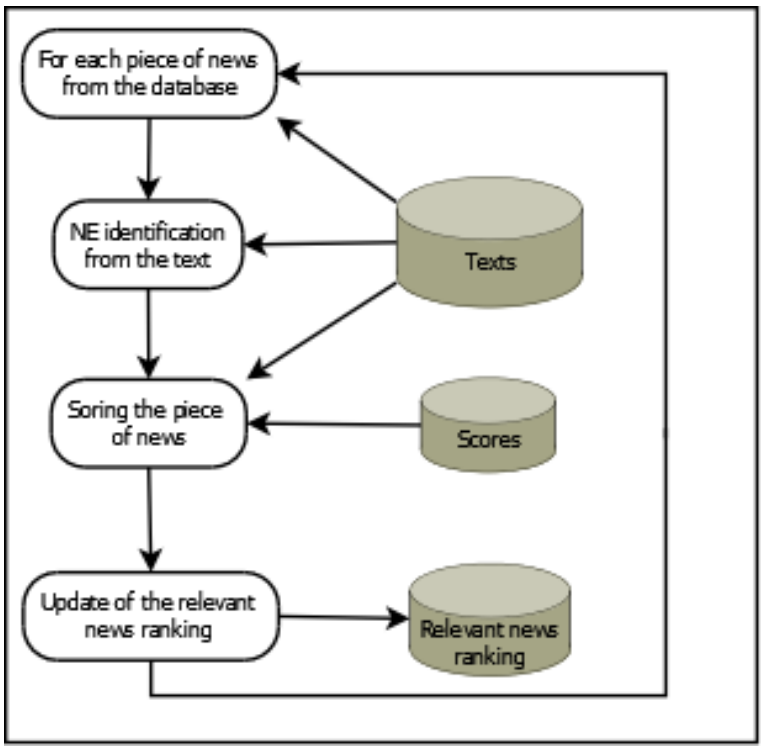

Fig. 3. Tagging close-up photo process.

Secondly, we have consider specific weights that assign a score for each range of occurrences of each name. The process selects news with higher score. To adjust the results have been tested with different weights, and determine which give us the desired news. The score is stored in the database, so each stored text corresponds to a set of characters and its score.

The relevance assessment process is launched daily by the staff to keep updated the documentation database.

3 Developed prototype

The developed system consists of an application designed to integrate into the desktop. This chapter describes the general architecture of biographical sheet generator which is shown in Fig. 4.

In order to obtain a biographical sheet, the journalist or documentalist user connects by a user interface (Fig. 5) to perform searches over both repositories: EMMA and Internet.

First, the system first fulfils a preliminary search to determine the presence of the character in the list of Named Entities in the documentary archive and

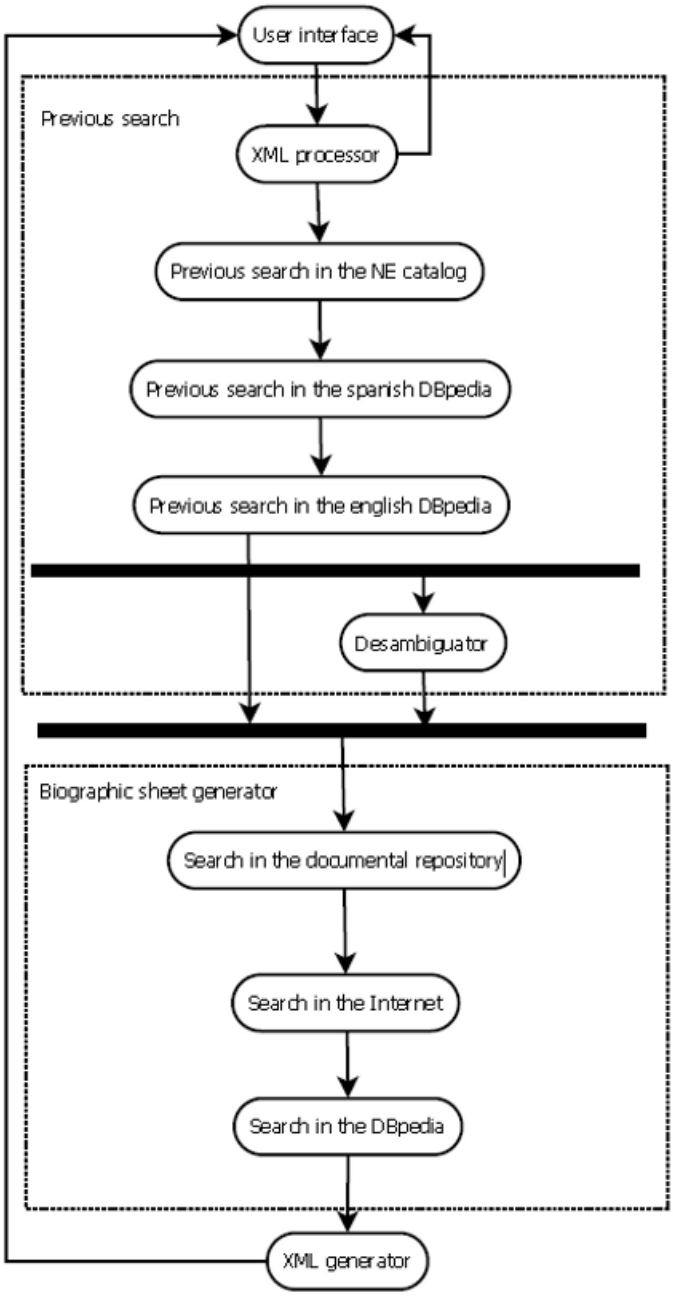

Fig. 4. General process of the biographical sheet generator.

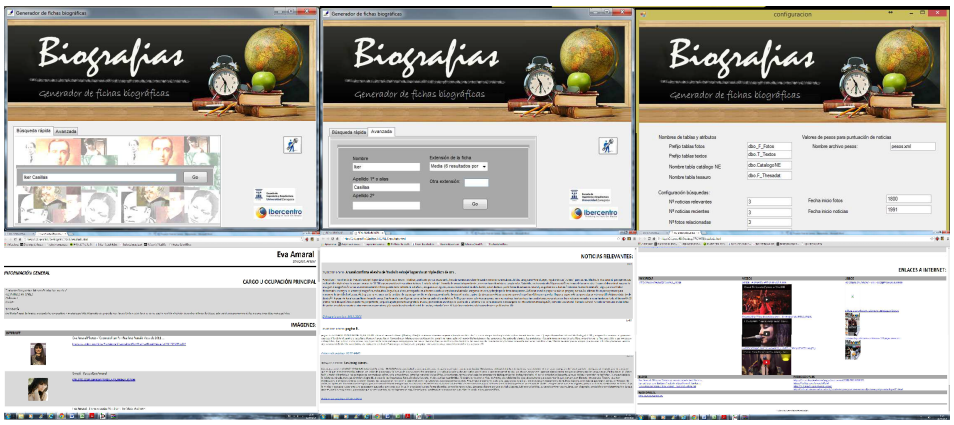

Fig. 5. Some screen-shots of the application.

in the DBpedia. If it do not appears, the character is discarded. If its presence is relevant (because it appears in both), and there are problems of ambiguity (coincidence of two or more entities with the same name), the system solves it.

With a character name set after the first stage, the process carries out a search to finding all the information that will be part of the biographical sheet. First, it searches in the documentary archive (news, photos), then Internet (biographical information on Wikipedia links to blogs, websites, books, other news, videos, social networks), and finally, it completes it with other biographical information by using DBpedia.

With this information the system generates an XML file that is eventually replaced by an HTML for its presentation to the user. These two steps are needed because the XML file gives a result that is independent of whether it is going to be shown on a single computer with a display, or integrated into a web platform.

We have had to make some essential preparatory work to achieve results in real time: to perform a previous search of each character, in order to extract close-up photos, and in order to filter and to sort the most relevant news or photographs stored in EMMA.

4 Case study

For the overall assessment of the results of the tool, we have conducted a small survey in a group of 100 workers of the Herald Group. Of these 20 belong to the Department of Documentation and on the other hand, 80 are journalists or experienced users in documentation tasks.

In this survey the biographical generator tool was presented and people are questioned about the usefulness in generating character sheets automatically, and the ability to simultaneously collect data from internal storage to the newspaper and the Internet. After testing the tool, a score of 1-10 on the importance given

to the development of their work (Relev), the utility (Util), the usability of the interface (Man) and finally they were asked about the novelty (New) among the tools commonly used. The result is shown in the following table:

Table 1. Survey among the professional users

Feature	**Relev**	**Util**	**Man**	**New**	**Total**
In/Out Searches	9	10	9	8	9
Autom. Dossiers	9	9	7	10	8.75

Table 2. Survey among the final users

Feature	**Relev**	**Util**	**Man**	**New**	**Total**
In/Out Searches	8	9	8	9	8.5
Autom. Dossiers	9	9	9	10	9.25

In the results we can see that the creation of sheets is a bit most valued by journalists, while collecting information simultaneously EMMA and Internet is slightly most appreciated by the documentalists. But in both departments the application is welcome.

Assessing the significance of the sheets is a very subjective task and entail reading all the news on a particular character. For example, we have found 52,000 news stored in the Heraldo de Aragón talking about Ramón y Cajal. So it is very complex to exactly evaluate the correctness of the search and sort algorithms. Therefore the evaluation of the tool was done empirically in consensus with the department documentation by choosing and weighting the elements that we can provide data on the value of an item (be cover, be odd page, number of words, etc.) . Also the evaluation of the results was made jointly by adjusting these weights for each element until the documentalist were satisfied with the results.

Regarding the evaluation of the labeling of the close-up photographs, we have obtained between 60 and 70% correct for photographs of any kind, and more than 90% if only photographs of people are chosen.

5 Related Work

We have search for other existing solutions to this problem. We have found the works described below:

– **http://omnibiography.com/** - Generates a biography but indeed you have to provide all the content, because it produces only a template.

- **http://www.vizify.com/** - Vizify is an on-line application to create a biography based on Social Media, more precisely in specific social networks like Facebook, Twitter, Foursquare and LinkedIn. This information is stored, reordered and displayed as a Mind Map, which becomes a kind of overview of ourselves. Really it is not very close from our solution.
- **http://www.biographyonline.net/** This tool owns pre-process sheets with biography, photos and links to related Web pages. You can not choose a character, because you only pick one that gives you the index.

Regarding applications that mix data stored in a private database with Internet information in the context of a media company, the authors have not found anything related.

6 Conclusions and Future Work

The main objective of this project was to develop a system able to generate the information sheet of a character automatically from a set of semi-structured texts of a media company, mixed with the data that can be extracted from Internet. The software application would be responsible for generating electronic documents containing information related to the selected character consisting of a selection of suitable news and relevant photographs, biographical information, links to websites, videos, social networks or links to other relevant information appearing on the Web.

At the end, we have implemented a tool that provides both documentalists and journalists a biographical profile of a character with complete and useful information from both enviroments: the internal database and the Web. The system significantly reduces the time to locate information and facilitate their work.

Finally, the main contributions made by this project are:

- To apply on a real case of study techniques of information extraction and natural language processing in order to find ways to extract the information useful to generate biographies from newspaper articles.
- To solve conflicts related to the ambiguity of similar names by studying possible algorithms and techniques to disambiguate them.
- To integrate in a information retrieval system techniques from information extraction to obtain data sheets, working at a time the search on a private repository with a search on the Web, and finally blending the results of the aforementioned searches.
- This study has also made a practical evaluation of the Semantic Web, using a semantic repository like DBpedia to retrieve concrete information.

Regarding disambiguation, is an open problem where we have approximated a solution for our specific case, but of course it can be improved. With respect the close-up photos, we have added functionality that does not usually exist in a media environment, but of course it can be enhanced in order to achieve better performance results.

Acknowledgment

This research work has been supported by the CICYT project TIN2010-21387-C02-02 and DGA-FSE. Thanks to Heraldo Group.

References

1. Garrido, A.L., Gomez, O., Ilarri, S., Mena, E.: Nass: news annotation semantic system. In: 23rd International Conference on Tools with Artificial Intelligence, IEEE (2011) 904–905
2. Garrido, A.L., Buey, M.G., Escudero, S., Peiro, A., Ilarri, S., Mena, E.: The GENIE project - a semantic pipeline for automatic document categorisation. In: 10th International Conference on Web Information Systems and Technologies, SCITEPRESS (2014)
3. Garrido, A.L., Buey, M.G., Ilarri, S., Mena, E.: GEO-NASS: A semantic tagging experience from geographical data on the media. In: 17th East-European Conference on Advances in Databases and Information Systems (ADBIS 2013), Genoa (Italy). Volume 8133., Springer (September 2013) 56–69
4. Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R., Ives, Z.: Dbpedia: A nucleus for a web of open data. Springer (2007)
5. Sekine, S., Ranchhod, E.: Named Entities: Recognition, Classification and Use. John Benjamins (2009)
6. Salton, G., Buckley, C.: Term-weighting approaches in automatic text retrieval. Information Processing and Management **24**(5) (1988) 513–523
7. Carreras, X., Chao, I., Padró, L., Padró, M.: FreeLing: An open-source suite of language analyzers. In: Fourth International Conference on Language Resources and Evaluation, European Language Resources Association (2004) 239–242
8. Astler, D., Chau, H., Hsu, K., Hua, A., Kannan, A., Lei, L., Nathanson, M., Paryavi, E., Rosen, M., Unno, H., et al.: Increased accessibility to nonverbal communication through facial and expression recognition technologies for blind/visually impaired subjects. In: The proceedings of the 13th international ACM SIGACCESS conference on Computers and accessibility, ACM (2011) 259–260
9. Tan, P.Y., Ibrahim, H., Bhargav, Y., Moorthy, P.S., Bhargav, Y., Moorthy, P.S., Kumar, J.N., Reddy, M.J., Anusha, T., Rao, N.S., et al.: Implementation of band pass filter for homomorphic filtering technique. International Journal of Computer Science **1**(5) (2013)
10. Luhn, H.P.: Auto-encoding of documents for information retrieval systems. IBM Research Center (1958)